



# SEMI E5-0709

## SEMI EQUIPMENT COMMUNICATIONS STANDARD 2 MESSAGE CONTENT (SECS-II)

This standard was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on May 13, 2009. It was available at [www.semi.org](http://www.semi.org) in June 2009 and on CD-ROM in July 2009. Originally published 1982; previously published November 2008.

**NOTICE:** The user's attention is called to the possibility that some implementations of this standard, particularly those related to the use of Stream 4, may involve the use of inventions covered by U.S. patents 4,884,674 and 5,216,613, and by other patents issued or pending, held by Texas Instruments Incorporated. By publication of this standard, SEMI takes no position respecting either the applicability or the validity of these or other patent rights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights and the risk of infringement of such rights, are entirely their own responsibility.

### Table of Contents

1 Purpose .....	3
2 Scope .....	3
3 Limitations .....	3
4 Referenced Standards and Documents .....	3
5 Terminology .....	4
6 The Message Transfer Protocol .....	5
6.1 <i>Intent</i> .....	5
6.2 <i>Message</i> .....	5
6.3 <i>Blocking Requirements</i> .....	5
6.4 <i>Message Header</i> .....	5
6.5 <i>Transaction Timeout</i> .....	5
6.6 <i>Multiple Open Transactions</i> .....	6
7 Streams and Functions .....	6
7.1 <i>Streams</i> .....	6
7.2 <i>Functions</i> .....	6
7.3 <i>Stream and Function Allocation</i> .....	6
8 Transaction and Conversation Protocols .....	7
8.1 <i>Intent</i> .....	7
8.2 <i>Transaction Definition</i> .....	7
8.3 <i>Transaction Level Requirements</i> .....	7
8.4 <i>Conversation Protocols</i> .....	7
9 Data Structures .....	8
9.1 <i>Intent</i> .....	8
9.2 <i>Item</i> .....	8
9.3 <i>List</i> .....	9
9.4 <i>Localized Character String Items</i> .....	9
9.5 <i>Example Data Structures</i> .....	10
9.6 <i>Data Item Dictionary</i> .....	11
9.7 <i>Variable Item Dictionary</i> .....	53
9.8 <i>Object Dictionary</i> .....	62
10 Message Detail .....	72
10.1 <i>Intent</i> .....	72
10.3 <i>Message Usage</i> .....	73
10.4 <i>Stream 0 and Function 0</i> .....	73
10.5 <i>Stream 1 Equipment Status</i> .....	73
10.6 <i>Stream 2 Equipment Control and Diagnostics</i> .....	80



10.7 Stream 3 Materials Status .....	99
10.8 Stream 4 Material Control .....	113
10.9 Stream 5 Exception Handling .....	127
10.10 Stream 6 Data Collection .....	132
10.11 Stream 7 Process Program Management .....	142
10.12 Stream 8 Control Program Transfer .....	159
10.13 Stream 9 System Errors .....	160
10.14 Stream 10 Terminal Services .....	163
10.15 Stream 11 Host File Services (Deleted) .....	165
10.16 Stream 12 Wafer Mapping .....	166
10.17 Stream 13 Data Set Transfers .....	178
10.18 Stream 14 Object Services .....	190
10.19 Stream 15 Recipe Management .....	210
10.20 Stream 16 Processing Management .....	238
10.21 Stream 17 Equipment Control and Diagnostics .....	254
10.22 Stream 18 Subsystem Control and Data .....	260
10.23 Stream 19 Recipe and Parameter Management .....	265
11 Message Documentation .....	275
11.1 Intent .....	275
11.2 Standard Form SECS-II Document .....	275
12 Units of Measure .....	276
12.1 Intent .....	276
12.2 Units Symbols .....	276
12.3 Compliance .....	277
12.4 SECS-II Units of Measure Identifiers .....	278
RELATED INFORMATION 1 .....	284
R1-1 The General Node Transaction Protocol .....	284
R1-2 Some Suggested Message Usage .....	284
R1-3 Notes on SECS-II Data Transfers .....	284
R1-4 Process Programs .....	287
R1-5 Suggested Baseline SECS Equipment Implementation .....	291

## 1 Purpose

1.1 The SEMI Equipment Communications Standard Part 2 (SECS-II) defines the details of the interpretation of messages exchanged between intelligent equipment and a host. This specification has been developed in cooperation with the Japan Electronic Industry Development Association Committee 12 on Equipment Communications.

1.1.1 It is the intent of this standard to be fully compatible with SEMI E4 — Equipment Communications Standard (SECS-I). It is also the intent to allow for compatibility with alternative message transfer protocols. The details of the message transfer protocol requirements are contained in § 6.

1.1.2 It is the intent of this standard to define messages to such a level of detail that some consistent host software may be constructed with only minimal knowledge of individual equipment. The equipment, in turn, may be constructed with only minimal knowledge of the host.

1.1.3 The messages defined in the standard support the most typical activities required for IC manufacturing. The standard also provides for the definition of equipment-specific messages to support those activities not covered by the standard messages. While certain activities can be handled by common software in the host, it is expected that equipment-specific host software may be required to support the full capabilities of the equipment.

## 2 Scope

2.1 SECS-II gives form and meaning to messages exchanged between equipment and host using a message transfer protocol, such as SECS-I.

2.1.1 SECS-II defines the method of conveying information between equipment and host in the form of messages. These messages are organized into categories of activities, called streams, which contain specific messages, called functions. A request for information and the corresponding data transmission is an example of such an activity.

2.1.2 SECS-II defines the structure of messages into entities called items and lists of items. This structure allows for a self-describing data format to guarantee proper interpretation of the message.

2.1.3 The interchange of messages is governed by a set of rules for handling messages called the transaction protocol. The transaction protocol places some minimum requirements on any SECS-II implementation.

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

## 3 Limitations

3.1 SECS-II applies to equipment and hosts used in the manufacturing of semiconductor devices. Examples of the activities supported by the standard are: transfer of control programs, material movement information, measurement data, summarized test data, and alarms.

3.1.1 The minimum compliance to this standard involves meeting the few constraints outlined in § 8. It is expected that a given piece of equipment will require only a subset of the functions described in this standard. The number of functions and the selection of functions will depend upon the equipment capabilities and requirements. For each piece of equipment, the exact format for each function provided must be documented according to the form outlined in § 10.

3.1.2 It is assumed that the equipment will define the messages used in a particular implementation of SECS-II. It is assumed the host will support equipment implementation.

## 4 Referenced Standards and Documents

### 4.1 SEMI Standards

SEMI E4 — SEMI Equipment Communications Standard 1 Message Transfer (SECS-I)

SEMI E6 — Guide for Semiconductor Equipment Installation Documentation

SEMI E148 — Specification for Time Synchronization and Definition of the TS-Clock Object

## 4.2 ANSI Standard<sup>1</sup>

ANSI X3.4-1977 — Code for Information Interchange (ASCII)

## 4.3 IEEE Standard<sup>2</sup>

IEEE 754 — Standard for Binary Floating Point Arithmetic

4.4 The Japan Electronic Industry Development Association (JEIDA) has requested that the SECS-II standard incorporate support for the JIS-8 codes for data exchange. This code would allow support for katakana characters in Japanese implementations of SECS-II.

JIS-6226 — JIS 8-bit Coded Character Set for Information Interchange, Japanese Industrial Standards.<sup>3</sup>

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

## 5 Terminology

### 5.1 Definitions

5.1.1 The following brief definitions refer to sections providing further information.

5.1.2 *block* — a physical division of a message used by the message transfer protocol (see § 6.3).

5.1.3 *conversation* — a sequence of related messages (see § 8.4).

5.1.4 *conversation timeout* — an indication that a conversation has not completed properly (see § 8.4.1).

5.1.5 *device ID* — a number between 0 and 32767 used in identifying the particular piece of equipment communicating with a host (see § 6.4.1).

5.1.6 *equipment* — the intelligent system which communicates with a host.

5.1.7 *function* — a specific message for a specific activity within a stream (see § 7.2).

5.1.8 *host* — the intelligent system which communicates with the equipment.

5.1.9 *interpreter* — the system that interprets a primary message and generates a reply when requested (see § 6.2).

5.1.10 *item* — a data element within a message (see § 9.2).

5.1.11 *item format* — a code used to identify the data type of an item (see § 9.2).

5.1.12 *list* — a group of items (see § 9.3).

5.1.13 *message* — a complete unit of communication (see § 6.2).

5.1.14 *message header* — information about the message passed by the message transfer protocol (see § 6.4).

5.1.15 *multi-block message* — a message sent in more than one block by the message transfer protocol (see § 6.3.2).

5.1.16 *originator* — the creator of a primary message (see § 6.2).

5.1.17 *packet* — a physical division of a message used by the message transfer protocol (see § 6.3).

5.1.18 *primary message* — an odd numbered message. Also, the first message of a transaction (see § 6.2 and § 7.2).

5.1.19 *reply* — the particular secondary message corresponding to a primary message (see § 6.2 and § 7.2).

5.1.20 *secondary message* — an even-numbered message. Also the second message of a transaction (see § 6.2 and § 7.2).

---

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

2 Institute of Electrical and Electronics Engineers, IEEE Operations Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, New Jersey 08855-1331, USA. Telephone: 732.981.0060; Fax: 732.981.1721; <http://www.ieee.org>

3 Japanese Industrial Standards. Available through the Japanese Standards Association, 1-24, Akasaka 4-Chome, Minato-ku, Tokyo 107-8440, Japan. Telephone: 81.3.3583.8005; Fax: 81.3.3586.2014; <http://www.jsa.or.jp>

5.1.21 *single-block message* — a message sent in one block by the message transfer protocol (see § 6.3.1).

5.1.22 *stream* — a category of messages (see § 7.1).

5.1.23 *transaction* — a primary message and its associated secondary message, if any (see § 8.2).

5.1.24 *transaction timeout* — an indication from the message transfer protocol that a transaction has not completed properly (see § 6.5).

## 6 The Message Transfer Protocol

6.1 *Intent* — SECS-II is fully compatible with the message transfer protocol defined by SECS-I. It is the intent of this standard to allow for compatibility with alternative message transfer protocols. The purpose of this section is to define the requirements of the interaction between an application using SECS-II and the message transfer protocol. The methods used to implement these requirements are not covered as a part of this standard. The terms used in this standard are those used by SECS-I. Equivalent terms may be different for other message transfer protocols.

6.2 *Messages* — The message transfer protocol is used to send messages between equipment and host. The message transfer protocol must be capable of sending a primary message, indicating whether a reply is requested; and, if a reply is requested, it must be capable of associating the corresponding secondary message or reply message with the original primary message. The term originator will refer to the creator of the original primary message. The term interpreter will refer to the entity that interprets the primary message at its destination and generates a reply when requested.

6.3 *Blocking Requirements* — The message transfer protocol must support the following SECS-II message blocking requirements.

6.3.1 *Single-Block Messages* — SECS-II requires that certain messages be sent in a single block or single packet by the message transfer protocol. Those messages defined in this standard as single-block SECS-II messages must be sent in a single-block or packet. The method used by the application software to tell the message transfer protocol that a particular message must be sent as a single block is not covered as part of this standard. For compatibility with SECS-I, the maximum length allowed for a single-block SECS-II message is 244 bytes. The minimum requirement for the message transfer protocol is to be able to send single-block SECS-II messages.

6.3.2 *Multi-Block Messages* — For compatibility with SECS-I, SECS-II messages that are longer than 244 bytes are referred to as multi-block messages. Also, certain SECS-II messages are allowed to be multi-block messages even if they otherwise meet the single-block length requirements. Certain older implementations may impose application-specific requirements on block sizes for certain incoming messages. Beginning with the 1988 revision of the standard, new applications may not impose application-specific requirements on incoming block sizes. Applications implemented before 1988 may impose such requirements.

6.4 *Message Header* — The message transfer protocol must provide the following information, called the message header, with every message. Only the content of the message header is defined by this standard. The exact format of the message header passed between the application and the message transfer protocol is not covered as part of this standard.

NOTE 1: In SECS-I, this information is contained in the 10 byte header of each block of a message.

6.4.1 *Device ID* — The message transfer protocol must be capable of identifying the device ID (0–32767) which indicates the source or destination of a message.

6.4.2 *Stream and Function* — The message transfer protocol must be capable of identifying to SECS-II a minimum 15 bit message identification code. In SECS-II, messages are identified by a stream code (0–127, 7 bits) and a function code (0–255, 8 bits). Each combination of stream and function represents a distinct message identification.

6.4.3 *Reply Requested* — The message transfer protocol must be capable of identifying whether a reply is requested to a primary message.

6.5 *Transaction Timeout* — It is presumed that the message transfer protocol will notify SECS-II in the event of failure to receive an expected reply message within a specified transaction timeout period.

NOTE 2: In SECS-I, a transaction timeout occurs if either the reply timeout (T3) is exceeded before the first block of a reply message is received or if the interblock timeout (T4) is exceeded before an expected block of a multi-block message is received.

6.6 *Multiple Open Transactions* — This standard allows, but does not require, the support of more than one concurrent open transaction.

## 7 Streams and Functions

7.1 *Streams* — A stream is a category of messages intended to support similar or related activities.

7.2 *Functions* — A function is a specific message for a specific activity within a stream. All the functions used in SECS-II will follow a numbering convention corresponding to primary and secondary message pairs. All primary messages will be given an odd-numbered function code. The reply message function code is determined by adding one to the primary message function code. The even-numbered function following a primary message which requests no reply is reserved and is not to be used. Function code 0 is reserved in all streams for aborting transactions as described in § 10.4.

7.3 *Stream and Function Allocation* — Some of the stream and function code combinations are reserved for this standard, while others are available for user definition. The stream and function codes reserved for this standard are as follows:

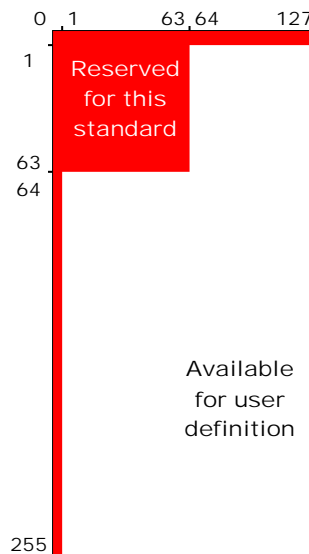
- In Stream 0, Functions 0–255.
- In Streams 1–63, Functions 0–63.
- In Streams 64–127, Function 0.

7.3.1 The stream and function codes available for user definition are as follows:

- In Streams 1–63, Functions 64–255.
- In Streams 64–127, Functions 1–255.

7.3.2 The stream and function code assignment can also be represented by the diagram shown in Figure 1.

7.3.3 The reserved codes assigned by this standard are listed in § 10. It is recognized that there will be user needs beyond the specific definitions given in this standard. In these situations, the streams and functions reserved for user definition should be used subject to the guidelines for minimum compliance outlined in § 8.



**Figure 1**  
**Stream and Function Allocation**

## 8 Transaction and Conversation Protocols

8.1 *Intent* — For an implementation to be in compliance with SECS-II, it must meet the minimum transaction requirements outlined in this section. The conversation protocols serve to further define the use and interaction between transactions.

8.2 *Transaction Definition* — A transaction forms the basis for all information exchanges in SECS-II. A transaction consists of either a primary message for which no reply is requested, or a primary message which requests a reply together with its corresponding secondary message. Secondary messages cannot request a reply.

8.3 *Transaction Level Requirements* — The following are the requirements to comply with the SECS-II protocol at the transaction level:

1. Respond to S1,F1 with S1,F2 as described in § 10.5.
2. For any received message that cannot be processed by the equipment, send the appropriate error message on Stream 9. As described in § 10.13, S9,F1, F3, F5, F7, or F11 are possible.
3. Format any other supported messages according to § 10.
4. Upon detection of a transaction timeout at the equipment, send S9F9 to the host.
5. Upon receipt of function 0 as a reply to a primary message, terminate the related transaction. No error message should be sent to the host by the equipment.

8.4 *Conversation Protocols* — A conversation is a series of one or more related transactions used to complete a specific task. A conversation should include all transactions necessary to accomplish the task and leave both the originator and interpreter free of resource commitments at its conclusion.

8.4.1 *Conversation Timeout* — A conversation timeout is used to indicate that a conversation has not completed properly. A conversation timeout is application-dependent, and the methods used for detecting conversation timeouts are not covered as part of this standard. A conversation timeout will terminate further action on the conversation, and will allow for the clearing of any committed resources. Upon detection of a conversation timeout at the equipment, S9,F13 should be sent to the host.

8.4.2 *Types of Conversations* — There are seven types of conversations which characterize all information exchanges in SECS-II:

1. A primary message with no reply is the simplest conversation. This message must be a single-block SECS-II message. The originator must assume that the interpreter responds to the message. This conversation is used where the originator can do nothing if the message is rejected.
2. If the interpreter has data that the originator wants, the data are requested with a primary message and the data returned to the originator as a reply message. It is assumed that the originator requesting the data is prepared to receive the amount of data returned. This is the request/data conversation.
3. If the originator wishes to send data in a single-block SECS-II message to the interpreter, then the originator sends the data and expects an acknowledgment from the interpreter. This is the send/acknowledge conversation.
4. If the originator has a multi-block SECS-II message to send for a particular exchange, then the originator must receive permission from the interpreter prior to sending the data. The first transaction requests permission to send, and the interpreter either grants or denies permission. If permission is granted, the originator sends the data and the interpreter replies appropriately. This is the inquire/grant/send/acknowledge conversation. Between the inquire and the send, the interpreter may commit some resources in preparation for the data. Consequently, a conversation timeout may be set by the interpreter at a time dependent upon the application, at which time the interpreter will free its resources and send an S9,F13 error message to the originator. Note that under the definition of S9,F13 in this standard, only the equipment should generate an error message to the host under these conditions.
5. There is a conversation related to the transfer of unformatted data sets between equipment and host. This conversation is described in detail in Stream 13 (see § 10.17).
6. There is a conversation related to the handling of material between equipment. This conversation is described in detail in Stream 4 (see § 10.8).
7. The originator may request information from the interpreter which requires some time to obtain (e.g., operator input is required). The first transaction requests the information and the interpreter responds in one of three

ways: (1) the information is returned, (2) the interpreter indicates that the information cannot or will not be obtained, or (3) the interpreter indicates that the information will be obtained and returned in a subsequent transaction, as specified for this conversation. For case number 3, the interpreter will initiate the subsequent transaction when the information is available.

8.4.2.1 Case 3 is the request/acknowledge/send/acknowledge transaction.

8.4.2.2 The originator of the request/acknowledge/send/acknowledge conversation may commit some resources in anticipation of the send/acknowledge transaction. Consequently, a conversation timeout may be set by the originator at a time dependent on the application. On timeout, the originator will free its resources and restart the conversation with the 'request', or send an S9,F13 error message. Note that under the definition of S9,F13 in this standard, only the equipment should generate an error message to the host under these conditions.

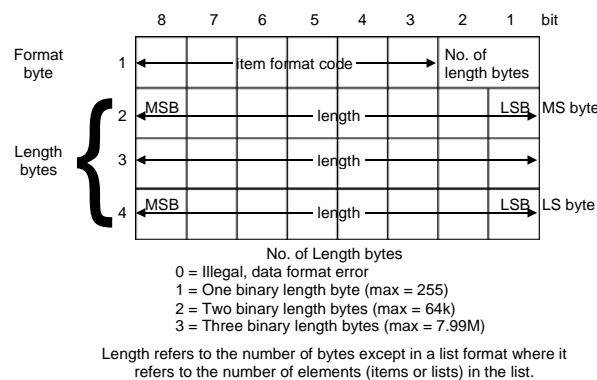
8.4.3 The key words, request, data, send, acknowledge, inquire, and grant are used in the function names as an aid to understanding the relationship between the messages and the conversation. Single message transactions do not use these words.

## 9 Data Structures

9.1 *Intent* — All information transmitted according to this standard will be formatted using two data structures, items and lists. These data structures define the logical divisions of the message, as distinct from the physical divisions of the message transfer protocol. They are intended to provide a self-describing internal structure to messages passed between equipment and host.

9.2 *Item* — An item is an information packet which has a length and format defined by the first 2, 3, or 4 bytes of the item. These first bytes are called the item header (IH). The item header consists of the format byte and the length byte(s) as shown in Figure 2. Bits one and two of the item header tell how many of the following bytes refer to the length of the item. This feature allows for long items without requiring the byte overhead for shorter items. The item length refers to the number of bytes following the item header, called the item body (IB), which is the actual data of the item. The item length refers only to the item body not including the item header, so the actual number of bytes in the message for one item is the item length plus 2, 3, or 4 bytes for the item header. All bytes in the item body are in the format specified in the format byte.

9.2.1 A zero-length in the format byte is illegal and produces an error. A zero-length in the item length bytes has a special meaning as defined in the detailed message definitions.



**Figure 2**  
**Item and List Header**

9.2.2 Bits 3–8 of the format byte of the item header define the format of the data which follows. Of the 64 possible formats, 16 are defined as shown in Table 1. Format code 0 is called a list and is defined in § 9.3. Format code 22 (octal) is called a localized string and is defined in § 9.4. The remaining 14 item formats define unspecified binary, code 10 (octal); Boolean, code 11 (octal); ASCII character strings, code 20 (octal); JIS-8 character strings, code 21 (octal) signed integer, codes 30, 31, 32, 34 (octal); floating point, codes 40, 44 (octal); and unsigned integer, codes 50, 51, 52, 54 (octal). These formats are used for groups of data which have the same representation in order to save



repeated item headers. In signed integers, negative values will be two's complement values. Floating point numbers will conform to the IEEE standard 754. Boolean values will be byte quantities, with zero being equivalent to false, and non-zero being equivalent to true.

**9.3 List** — A list is an ordered set of elements, where an element can be either an item (§ 9.2) or a list. The list header (LH) has the same form as an item header with format type 0. However, the length bytes refer to the number of elements in the list rather than to the number of bytes. The list structure allows grouping items of related information which may have different formats into a useful structure.

**9.3.1** A zero-length in the format byte is illegal and produces an error. A zero-length in the list length bytes has a special meaning, which is defined in the detailed message definitions.

**Table 1 Item Format Codes**

<i>Format Code (Bit 876543)</i>		<i>Meaning</i>
<i>Binary</i>	<i>Octal</i>	<i>Data after the heading has the following form</i>
000000	00	LIST (length in elements)
001000	10	Binary
001001	11	Boolean
010000	20	ASCII <sup>#1</sup>
010001	21	JIS-8
010010	22	2 byte character <sup>#2,#4</sup>
011000	30	8 byte integer (signed) <sup>#2</sup>
011001	31	1 byte integer (signed)
011010	32	2 byte integer (signed) <sup>#2</sup>
011100	34	4 byte integer (signed) <sup>#2</sup>
100000	40	8 byte floating point <sup>#3</sup>
100100	44	4 byte floating point <sup>#3</sup>
101000	50	8 byte integer (unsigned) <sup>#2</sup>
101001	51	1 byte integer (unsigned)
101010	52	2 byte integer (unsigned) <sup>#2</sup>
101100	54	4 byte integer (unsigned) <sup>#2</sup>

<sup>#1</sup> Non-printing characters are equipment-specific.

<sup>#2</sup> Most significant byte sent first.

<sup>#3</sup> IEEE 754. The byte containing the sign bit is sent first.

<sup>#4</sup> The code for Multi-byte character must be specified in the data in the first 2 bytes of the TEXT item.

<sup>#5</sup> Changes in integer format codes may conflict with earlier implementations.

**9.4 Localized Character String Items** — A localized character string is an item which is used for representing a string of multi-byte characters. Because there are many different encoding schemes and the information could be in any one of a number of languages, these characteristics must also be included in the item. Thus for localized character strings which use item format code 22 (octal), there is an additional localized string header (LSH).

**9.4.1** This localized string header follows the item header and precedes the string. The localized string header is part of the item data, thus the length of the header (2 bytes) is included in the length in the item header. The length of the localized string itself is the number of bytes that it occupies, regardless of the number of characters that represents the string. The localized string header followed by the string together comprise the localized string item. For example, a 2 byte localized string (which may represent a single character), because of the 2 byte length of the localized string header, will have a 4 byte length in the item header.

9.4.2 The LSH is a 16 bit number which specifies the encoding method used for the string. Defined values for the encoding are as follows:

**Table 2**

<i>Encoding Code (Decimal)</i>	<i>Encoding Scheme</i>	<i>Notes</i>
0	none	reserved
1	ISO 10646 UCS-2	Unicode 2.0
2	UTF-8	Transformation of ISO 10646 UCS-2
3	ISO 646-1991	ASCII, 7-bit
4	ISO 8859-1	ISO Latin-1, Western Europe
5	ISO 8859-11 (proposed)	Thai
6	TIS 620	Thai (will be supported by ISO 8859-11)
7	IS 13194 (1991)	ISCII
8	Shift JIS	
9	Japanese EUC-JP	
10	Korean EUC-KR	
11	Simplified Chinese GB	
12	Simplified Chinese EUC-CN	
13	Traditional Chinese Big5	
14	Traditional Chinese EUC-TW	

9.4.3 Encoding Codes from 15–32767 are reserved for future expansion. Encoding codes from 32768–65535 are available for custom purposes.

9.5 *Example Data Structures* — The data structures for different types of items are illustrated in the following examples:

- a. An item contains one binary code 10101010.

```

bit
87654321

00100001    Item, binary, 1 length byte
00000001    1 byte long
10101010    data byte

```

- b. An item contains three ASCII characters ABC.

```

01000001    Item ASCII, 1 length byte
00000011    Three bytes long
01000001    ASCII A
01000010    ASCII B
01000011    ASCII C

```

- c. An item contains three binary numbers in 2-byte signed integer form.

```

01101001    Item, 2-byte integers
00000110    6 bytes total (6/2 = 3 integers)
xxxxxxxx    MSByte number x
xxxxxxxx    LSByte number x
yyyyyyyy    MSByte number y
yyyyyyyy    LSByte number y
zzzzzzzz    MSByte number z
zzzzzzzz    LSByte number z

```



- d. An item contains one 4-byte IEEE floating point number.

```
10010001    Item, 4-byte floating point
00000100    4 bytes (4/4 = 1 number)
ffffffff
ffffffff    Floating point number
ffffffff    in IEEE 754
ffffffff
```

- e. A message is sent from device 66 telling the host that the temperature at point T1 has exceeded a preset process limit. The message ID is stream 5, function 1, and the data consists of a list of three items. The first item is a code for the alarm set and the alarm category code. The second item is the equipment-specific alarm number for this alarm (e.g., 17). The third item is a string of text giving a brief description of the alarm (e.g., "T1 HIGH"). No reply is requested. The complete message including the header is as follows:

```
10000000    R = 1 (to the host)
01000010    Device Code = 66
00000101    Stream 5, W = 0
00000001    Function 1
10000000    E = 1
00000001    Block 1
00000000
00000000    System bytes = 0
00000000
00000001    List
00000011    3 Elements
00100001    Binary Item next byte length
00000001    1 byte long
00000100    Alarm set, category 4
01100101    Item, 1-byte integer, next byte length
00000001    1 byte long
00010001    Alarm 17
01000001    Item, ASCII, next byte length
00000111    7 characters
01010100    ASCII T
00110001    ASCII 1
00100000    ASCII space
01001000    ASCII H
01001001    ASCII I
01000111    ASCII G
01001000    ASCII H
```

9.5.1 Using SEMI E4 SECS-I transmission, the entire message contains 1 byte length (not shown), 10 bytes of header, 17 bytes of data, and 2 bytes checksum (not shown) for a total of 30 bytes. At 9600 baud transmission, the message would be sent in 31 milliseconds.

9.6 *Data Item Dictionary* — This section defines the data items used in the standard SECS-II messages described in § 10, Message Detail.

**Name:** A unique mnemonic name for this data item. This name is used in message definitions.

**Format:** The allowable item format codes which can be used for this standard data item. Item format codes are shown in octal, as described in Table 1, Item Format Codes. The notation "3()" indicates any of the signed integer formats (30, 31, 32, 34). The notation "4()" indicates any of the floating point formats (40, 44). The notation "5()" indicates any of the unsigned integer formats (50, 51, 52, 54). The notation "0" indicates that a list with user-defined structure may be used. Where more than one format is shown, a given implementation can use any of the formats specified.

**Description:** A description of the data item, with the meanings of specific values.

**Where Used:** The standard messages in which this data item appears.

**Table 3 Data Item Dictionary**

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ABS	10	Any binary string.		S2F25, F26
ACCESSMODE	51	Load Port Access Mode. Possible values are:	0 = Manual 1 = Auto	S3F27
ACDS	32, 52	After Command Codes.	Vector of all command codes which defined command must succeed within the same block.	S7F22
ACKA	11	Indicates success of a request:	TRUE is successful else FALSE	S5F14, F15, F18; S16F2, F4, F6, F12, F14, F16, F18, F24, F26, F28, F30; S17F4, F8, F14
ACKC3	10	Acknowledge code, 1 byte.	0 = Accepted >0 = Error, not accepted 1-63 = Reserved	S3F6, F8, F10
ACKC5	10	Acknowledge code, 1 byte.	0 = Accepted >0 = Error, not accepted 1-63 = Reserved	S5F2, F4
ACKC6	10	Acknowledge code, 1 byte.	0 = Accepted >0 = Error, not accepted 1-63 = Reserved	S6F2, F4, F10, F12, F14
ACKC7	10	Acknowledge code, 1 byte.	0 = Accepted 1 = Permission not granted 2 = Length error 3 = Matrix overflow 4 = PPID not found 5 = Mode unsupported 6 = Command will be performed with completion signaled later >6 = Other error 7-63 = Reserved	S7F4, F12, F14, F16, F18, F24, F32; S7F38, F40, F42, F44
ACKC7A	31, 51	Acknowledge Code, 1 byte.	0 = Accepted 1 = MDLN is inconsistent 2 = SOFTREV is inconsistent 3 = Invalid CCODE 4 = Invalid PPARM value 5 = Other error (described by ERRW7) 6-63 = Reserved	S7F27
ACKC10	10	Acknowledge Code, 1 byte.	0 = Accepted for display 1 = Message will not be displayed 2 = Terminal not available 3-63 = Reserved	S10F2, F4, F6, F10

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ACKC13	10	Return code for secondary messages 1 byte.	0 = O.K. 1 = ERROR: Try Later 2 = ERROR: Unknown Data Set name 3 = ERROR: Illegal Checkpoint value 4 = ERROR: Too many open Data Sets 5 = ERROR: Data set open too many times 6 = ERROR: No open Data Set 7 = ERROR: Cannot continue 8 = ERROR: End of Data 9 = ERROR: Handle in Use >10 = ERROR: Pending Transaction 11-127 = Reserved	S13F2, F4, F6, F8
ACKC15	10	Return code for secondary messages, 1 byte.	0 = Accepted 1 = command will be performed with completion signaled later 2 = DSNAME not found 3 = Permission not granted 4 = other error 5-63 = Reserved	S15F50, F52
AGENT	20			S15F11, F12, F21, F22, F25
ALCD	10	Alarm code byte.	bit 8 = 1 means alarm set bit 8 = 0 means alarm cleared bit 7-1 is alarm category 0 = Not used 1 = Personal safety 2 = Equipment safety 3 = Parameter control warning 4 = Parameter control error 5 = Irrecoverable error 6 = Equipment status warning 7 = Attention flags 8 = Data integrity >8 = Other categories 9-63 = Reserved	S5F1, F6
ALED	10	Alarm enable/disable code, 1 byte.	bit 8 = 1 means enable alarm bit 8 = 0 means disable alarm	S5F3
ALID	3(), 5()	Alarm identification.		S5F1, F3, F5, F6
ALT_X	20	Alarm text limited to 120 characters.		S5F1, F6

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ATTRDATA	0, 10, 20, 3(), 4(), 5(), 11	Contains a specific attribute value for a specific object.		S1F20; S3F17, F18; S13F14, F16; S14F1, F2, F3, F4, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18; S18F2, F3
ATTRID	20, 5()	Identifier for an attribute for a specific type of object.		S1F19; S3F17, F18; S13F14, F16; S14F1, F2, F3, F4, F8, F9, F10, F11, F12, F13, F14, F15, F16, F17, F18; S18F1, F3
ATTRRELN	51	The relationship that a specified qualifying value has to the value of an attribute of an object instance (the value of interest):	0 = The qualifying value is <u>equal to</u> the value of interest, 1 = The qualifying value is not equal to the value of interest, 2 = The qualifying value is less than the value of interest, 3 = The qualifying value is less than or equal to the value of interest, 4 = The qualifying value is greater than the value of interest, 5 = The qualifying value is greater than or equal to the value of interest, 6 = The qualifying value is present (contained in the set of) the value of interest, 7 = The qualifying value is absent (not contained in the set of) the value of interest, >7 = Reserved.	S14F1
BCDS	32, 52	Before Command Codes.	Vector of all command codes which defined command must precede within the same block.	S7F22
BCEQU	20, 51	Bin code equivalents.	Array of all codes that are to be processed. Must be the same format as BINLT and NULBC. Zero length indicates all codes be sent.	S12F3, F4
BINLT	20, 51	The Bin List.	Is an array of bin values. Format must be the same as used in NULBC and BCEQU.	S12F7, F9, F11, F14, F16, F18

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
BLKDEF	31, 51	Block Definition.	Blocks define the range for before/after code checking (specified by BCDS, IBCDS, NBCDS, ACDS, IACDS, and NACDS). BLKDEF specifies whether the command being defined starts a block (+1), terminates a block (-1), or is within the body of a block (0). All other values are invalid. The outermost block of a process program is implicit, and is bounded by the first and last command of the process program. Before/after checks for a particular command are performed with all other commands in the same block and at the same nesting level. For the purpose of before/after checking, a set of commands making up a block is considered to be a single body command of its containing block. This command has the before/after restrictions of the start block command which begins the block.	S7F22
BPD	10	Boot program Data.		S8F2
BYTMAX	3(), 5()	Byte Maximum.	Maximum length of process program. A value of zero indicates no limit. Negative values are invalid.	S7F22
CAACK	51	Carrier Action Acknowledge Code, 1 byte.	0 = Acknowledge, command has been performed. 1 = Invalid command 2 = Can not perform now 3 = Invalid data or argument 4 = Acknowledge, request will be performed with completion signaled later by an event. 5 = Rejected. Invalid state. 6 = Command performed with errors. 7-63 = Reserved.	S3F18, F20, F22, F24, F26, F30, F32
CARRIERACTION	20	Specifies the action requested for a carrier.		S3F17
CARRIERID	20	The identifier of a carrier.		S3F17
CARRIERSPEC	20	The object specifier for a carrier. Conforms to OBJSPEC.		F29, F31

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
CATTRDATA	0, 10, 20, 3(), 4(), 5(), 11	The value of a carrier attribute.		S3F17
CATTRID	20	The name of a carrier attribute.		S3F17
CCODE	20, 32, 34, 52, 54	Command Code.	Each command code corresponds to a unique process operation the machine is capable of performing.	S7F22, F23, F26, F31, F39, F43
CEED	11	Collection event or trace enable/disable code, 1 byte.	FALSE = Disable TRUE = Enable	S2F37; S17F5
CEID	20, 3(), 5()	Collected event ID.		S2F35, F37; S6F3, F8, F9, F11, F13, F15, F17; S17F5, F9, F10, F11, F12
CEPACK	0, 51	Command Enhanced Parameter Acknowledge. If a specific value of CPNAME is defined to have a CEPVAL that is a LIST, then CEPACK shall have the same structure as the corresponding list format of CEPVAL as used in S2,F49. Otherwise CEPACK will be a 1 byte integer. Enumerated:	0 = No error 1 = Parameter name (CPNAME) does not exist 2 = Illegal value specified for CEPVAL 3 = Illegal format specified for CEPVAL 4 = Parameter name (CPNAME) not valid as used 5-63 = Reserved	S2F50
CEPVAL	0, 10, 11, 20, 21, 3(), 4(), 5()	Command Enhanced Parameter Value. A specific application of CEPVAL shall always be identified with a specific value of CPNAME. A CEPVAL has the following forms: a single (non-list) value (e.g., CPVAL), a list of single items of identical format and type, or a list of items of the form.	L, 2 1. CPNAME 2. CEPVAL	S2F49
CKPNT	54	Checkpoint as defined by the sending system.		S13F3, F6
CMDA	31, 51	Command acknowledge code.	0 = Completed or done 1 = Command does not exist 2 = Cannot perform now >2 = Other equipment-specific error 3-63 = Reserved	S2F22, F28
CMDMAX	3(), 5()	Command Maximum.	Maximum number of commands to be allowed in a process program. A value of zero indicates no limit. Negative values are invalid.	S7F22



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
CNAME	20	Command Name = 16 characters.	Text string unique among other CNAMEs in PCD which describes the processing done by the equipment for the corresponding CCODE.	S7F22
COLCT	5()	Column count in die increments.		S12F1, F4
COLHDR	20	Text description of contents of TBLELT. 1–20 characters.		S13F13, F16
COMMACK	10	Establish Communications Acknowledge Code, 1 byte.	0 = Accepted 1 = Denied, Try Again 2–63 Reserved	S1F14
COMPARISONOPERATOR	51	Choice of available operators that compare the supplied value to the current attribute value. Evaluated as “Current value XX supplied value” where XX is one of the enumerated values (e.g., “GT”).	0 = EQ - Equals (numeric or string) 1 = NotEQ - Not Equal (numeric or string) 2 = LT - Less Than (numeric) 3 = LE - Less than or equal to (numeric) 4 = GT - Greater Than (numeric) 5 = GE - Greater than or equal to (numeric) 6 = Like - Contains the substring (string) 7 = NotLike - Does not contain the substring (string) 8–255 Reserved	S19F1
CONDITION	20	Provides condition information for a subsystem component. Used in the data item in the CONDITIONLIST.		See CONDITION LIST
CONDITIONLIST	0	A list of CONDITION data sent in a fixed order. CONDITIONLIST has the following form:	L, s 1.<CONDITION1> . s.<CONDITIONs>	S18F16
CPACK	10	Command Parameter Acknowledge Code, 1 byte.	1 = Parameter Name (CPNAME) does not exist 2 = Illegal Value specified for CPVAL 3 = Illegal Format specified for CPVAL >3 = Other equipment-specific error 4–63 Reserved	S2F42
CPNAME	20, 3(), 5()	Command Parameter Name.		S2F41, F42, F49, F50; S4F21, F29; S16F5, F27
CPVAL	10, 11, 20, 21, 30, 5()	Command Parameter Value.		S2F41, F49; S4F21, F29; S16F5, F27; S18F13

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
CSAACK	10	Equipment Acknowledgement code, 1 byte.	0 = Everything correct 1 = Busy 2 = Invalid SPID 3 = Invalid data >3 = Equipment-specific error 4-63 Reserved	S2F8
CTLJOBCMD	51	Control Job command codes are assigned as follows:	1 = CJStart 2 = CJPause 3 = CJResume 4 = CJCancel 5 = CJDeselect 6 = CJStop 7 = CJAbort 8 = CJHOQ	S16F27
CTLJOBID	20	Identifier for Control Job. Conforms to OBJID.		S16F27
DATA	20	A vector or string of unformatted data.		S3F29, F31; S18F6, F7
DATAACK	10	Acknowledge code for data.	0 = Acknowledge 1 = Unknown DATAID 2 = At least parameter is invalid 3-63 Reserved	S14F22
DATAID	20, 3(), 5()	Data ID.		S2F33, F35, F39, F45, F49; S3F15, F17; S4F19, F25; S6F3, F5, F7, F8, F9, F11, F13, F16, F18, F27; S13F11, F13, F15; S14F19, F21, F23; S15F27, F29, F33, F35, F37, F39, F41, F43, F45, F47, F49; S16F1, F3, F5, F11, F13; S17F1, F5, F9
DATALength	3(), 5()	Total bytes to be sent.		S2F39; S3F15, F29, F31; S4F25; S6F5; S13F11; S14F23; S16F1, F11; S18F5, F7, S19F19
DATASEG	20	Used to identify the data requested.		S3F29, F31; S18F5, F7
DATASRC	20	Object type for Data Source Objects.		S14F1, F3, F6, F7, F8; S17F1

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
DATLC	51	Data location.	Location of invalid data, represented in bytes, measured from the start of the message in question, excluding all header bytes.	S12F19
DELRSPSTAT	51	Status response for the Delete PDE request.	0 = OK - Successfully deleted specified PDEs. 1 = PDEnotFound - PDE was not found on the service provider. 2 = PDElocked - The PDE is present, but cannot be deleted at this time. For example, the PDE might be executing or reserved for a job. The PDE was not deleted. 3-254 Reserved 255 = Other - A problem occurred that is not described by the other enumerated values. The PDE was not deleted.	S19F4
DIRRSPSTAT	51	Status response for the GET PDE Directory request.	0 = OK - Successfully returned requested data. 1 = BadFilter - One or more of the PDE filter specified were not properly specified. 2 = BadAttribute - One or more of the PDE attributes requested to be returned do not exist. 3 = DateTimeCompareFailed - The filtering of a time formatted field failed because the comparison was indeterminate for one or more PDE's. 4-254 Reserved 255 = Other - A problem occurred that is not described by the other enumerated values.	S19F2
DRACK	10	Define Report Acknowledge Code, 1 byte.	0 = Accept 1 = Denied. Insufficient space. 2 = Denied. Invalid format. 3 = Denied. At least one RPTID already defined. 4 = Denied. At least VID does not exist. >4 = Other errors 5-63 Reserved	S2F34
DSID	20, 3(), 5()	Data set ID.		S6F3, F8, F9
DSNAME	20	The name of the Data Set.	The minimum length is zero. The maximum is 200 characters.	S13F1, F2, F3, F4, S7F37, F39, F41, F43; S15F49, F51

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
DSPER	20	Data sample period. DSPER has two allowable formats:	<p>Format 1: hhmmss, 6 bytes</p> <p>Format 2: hhmmsscc, 8 bytes</p> <p>Where "hh" is hours, "mm" is minutes, "ss" is seconds' and "cc" is centiseconds.</p> <p>Equipment shall either (1) support only Format 1, or (2) support both Format 1 and Format 2. Equipment shall document which formats it accepts.</p> <p>Equipment which supports Format 2 need not necessarily support a minimum DSPER of 1 centisecond, nor a trace resolution of 1 centisecond, but equipment suppliers shall document its trace performance limits.</p>	S2F23
DUTMS	20	Die Units of Measure.	Use units description per SEMI E5 § 12.	S12F1, F4
DVNAME	3(), 20, 5()	Data value name.		S6F3, F8
DVVAL	0, 10, 11, 20, 21, 3(), 4(), 5()	Data value.		S6F3, F8, F9
EAC	10	Equipment acknowledge code, 1 byte.	<p>0 = Acknowledge</p> <p>1 = Denied. At least one constant does not exist</p> <p>2 = Denied. Busy</p> <p>3 = Denied. At least one constant out of range</p> <p>&gt;3 = Other equipment-specific error</p> <p>4-63 Reserved</p>	S2F16
ECDEF	10, 11, 20, 21, 3(), 4(), 5()	Equipment constant default value.		S2F30
ECID	3(), 20, 5()	Equipment Constant ID.		S2F13, F15, F29, F30
ECMAX	10, 11, 20, 21, 3(), 4(), 5()	Equipment constant maximum value.		S2F30
ECMIN	10, 11, 20, 21, 3(), 4(), 5()	Equipment constant minimum value.		S2F30
ECNAME	20	Equipment constant name.		S2F30

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ECV	10, 11, 20, 21, 3(), 4(), 5()	Equipment Constant Value.		S2F14, F15
EDID	10, 20, 3(), 5()	Expected data Identification.	Three possible responses. MEXP EDID EDID S02F03, <SPID> A[6] S03F13, <PTN> B[1] S07F03, <PPID> A[80], B[80]	S9F13
EMID	10, 20	Equivalent material ID (16 bytes maximum).		S3F9
EPD	10	Executive program data.		S8F4
EQNAME	20	A unique ASCII equipment identifier assigned by the factory to the equipment. Limited to a maximum of 80 characters.		S4F29
ERACK	10	Enable/Disable Event Report. Acknowledge Code, 1 byte.	0 = Accepted 1 = Denied. At least one CEID does not exist >1 = Other Errors 2-63 Reserved	S2F38
ERRCODE	5()	Code identifying an error.	0 = No error 1 = Unknown object in Object Specifier 2 = Unknown target object type 3 = Unknown object instance 4 = Unknown attribute name 5 = Read-only attribute - access denied 6 = Unknown object type 7 = Invalid attribute value 8 = Syntax error 9 = Verification error 10 = Validation error 11 = Object identifier in use 12 = Parameters improperly specified 13 = Insufficient parameters specified 14 = Unsupported option requested 15 = Busy 16 = Not available for processing 17 = Command not valid for current state 18 = No material altered 19 = Material partially processed 20 = All material processed	S1F20; S3F18, F30, F32; S4F20, F22, F23, F33, F35; S5F14, F15, F18; S13F14, F16; S14F2, F4, F6, F8, F10, F12, F14, F16, F18, F26, F28; S15F28, F18, F20, F22, F24, F26, F30, F32, F34, F36, F38, F40, F42, F44, F48, F53; S16F12, F14, F16, F18, F24, F26, F28; S17F2, F4, F6, F8, F10, F12, F14

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ERRCODE (con't)	5()	Code identifying an error.	21 = Recipe specification related error 22 = Failed during processing 23 = Failed while not processing 24 = Failed due to lack of material 25 = Job aborted 26 = Job stopped 27 = Job cancelled 28 = Cannot change selected recipe 29 = Unknown event 30 = Duplicate report ID 31 = Unknown data report 32 = Data report not linked 33 = Unknown trace report 34 = Duplicate trace ID 35 = Too many data reports 36 = Sample period out of range 37 = Group size too large 38 = Recovery action currently invalid 39 = Busy with another recovery currently unable to perform the recovery 40 = No active recovery action 41 = Exception recovery failed 42 = Exception recovery aborted 43 = Invalid table element 44 = Unknown table element 45 = Cannot delete predefined 46 = Invalid token 47 = Invalid parameter 48 = Load port does not exist 49 = Load port already in use 50 = Missing Carrier 51-63 = Reserved (data formats 51, 52, 54, or 50 must be used) 64-32767 = User defined (data formats 52, 54, or 50 must be used) 32768 = Action will be performed at earliest opportunity	S1F20; S3F16, F30, F32; S4F20, F22, F23, F33, F35; S5F14, F15, F18; S13F14, F16; S14F2, F4, F6, F8, F10, F12, F14, F16, F18, F26, F28; S15F28, F18, F20, F22, F24, F26, F30, F32, F34, F36, F38, F40, F42, F44, F48, F53; S16F12, F14, F16, F18, F24, F26, F28; S17F2, F4, F6, F8, F10, F12, F14

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ERRCODE (con't)	5()	Code identifying an error.	32769 = Action can not be performed now 32770 = Action failed due to errors 32771 = Invalid command 32772 = Client Alr 32773 = Duplicate ClientID 32774 = Invalid ClientType 32775 = IncompatibleVersions 32776 = Unrecognized ClientID (Client not currently connected) 32777 = Failed (Completed Unsuccess-fully) 32778 = Failed (Unsafe) – External intervention required 32779 = Sensor-Detected Obstacle 32780 = Material Not Sent 32781 = Material Not Received 32782 = Material Lost 32783 = Hardware Failure 32784 = Transfer Cancelled 32785-32792 reserved for future use by SEMI E127 service requests. 32793-65335 Reserved (data formats 52, 54, or 50 must be used) 65536 or above = User defined data formats 54 or 50 must be used)	S1F20; S3F16, F30, F32; S4F20, F22, F23, F33, F35; S5F14, F15, F18; S13F14, F16; S14F2, F4, F6, F8, F10, F12, F14, F16, F18, F26, F28; S15F28, F18, F20, F22, F24, F26, F30, F32, F34, F36, F38, F40, F42, F44, F48, F53; S16F12, F14, F16, F18, F24, F26, F28; S17F2, F4, F6, F8, F10, F12, F14
ERRTEXT	20	Text string describing the error noted in the corresponding ERRCODE. Limited to 120 characters maximum.		S1F20; S3F16, F18, F20, F22, F24, F26, F30, F32; S4F20, F22, F23, F33, F35; S5F14, F15, F18; S13F14, F16; S14F2, F4, F6, F8, F10, F12, F14, F16, F18, F26, F28; S15F28, F30, F32, F34, F36, F38, F40, F42, F44, F48, F53; S16F12, F14, F16, F18, F24, F26, F28; S17F4, F8, F14
ERRW7	20	Text string describing error found in process program.		S7F27

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
EVNTSRC	20	Object type for Event Source Objects.		S17F5, F9, F10, F11, F12
EXID	20	Unique identifier for the exception. Maximum length of 20 characters.		S5F9, F11, F13, F14, F15, F17, F18
EXMESSAGE	20	Text which describes the nature of the exception.		S5F9, F11
EXRECVRA	20	Text which specifies a recovery action for an exception. Maximum length of 40 bytes.		S5F9, F13
EXTYPE	20	Text which identifies the type of an exception. It is usually a single word of text.	"ALARM" "ERROR"	S5F9, F11; S14F1, F2, F8
FCNID	51	Function Identification.		S2F43, F44
FFROT	52	Film Frame Rotation.	In degrees from the bottom CW. (Bottom equals zero degrees.) Zero length indicates not used.	S12F1, F3
FILDAT	10, 20	Data from the Data Set.	The maximum length is the RECLEN from Open Data Set Data.	S13F6
FNLOC	52	Flat/Notch Location.	In degrees from the bottom CW. (Bottom equals zero degrees.) Zero length indicates not used.	S12F1, F3, F4
FRMLEN	3(), 5()	Formatted Process Program Length.	If greater than zero, indicates PPID is available as a formatted process program and its length in bytes. If zero, the PPID is not available as a formatted program. Negative values are invalid.	S7F34
GETRSPSTAT	51	Status response for the Get PDE and Get PDEheader requests.	0 = OK - PDE will be transferred. 1 = PDENotFound - The PDE is not available. 2 = PDELocked - The PDE exists, but cannot be transferred now. 3 = TransferNotAllowed - The PDE exists, but transfer is prohibited for this PDE or type of PDE. 4-254 Reserved 255 = Other - A problem occurred that is not described by the other enumerated values.	S19F6, S19F8



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
GRANT	10	Grant code, 1 byte.	0 = Permission Granted 1 = Busy, Try Again 2 = No Space Available 3 = Duplicate DATAID >3 = Equipment Specific Error Code 4-63 Reserved	S2F2, F40; S3F16; S4F26; S13F12; S14F24; S16F2, F12; S19F20
GRANT6	10	Permission to send, 1 byte.	0 = Permission granted 1 = Busy, try again 2 = Not interested >2 = Other errors 3-63 Reserved	S6F6
GRNT1	10	Grant code, 1 byte.	0 = Positive response, transfer ok 1 = Busy, try again 2 = No space 3 = Map too large 4 = Duplicate ID 5 = Material ID not found 6 = Unknown map format >6 = Error 7-63 Reserved	S12F6
HANDLE	3(), 5()	Logical unit or channel.		S13F3, F4, F5, F6, F7, F8
HCACK	10	Host Command Parameter Acknowledge Code, 1 byte.	0 = Acknowledge, command has been performed 1 = Command does not exist 2 = Cannot perform now 3 = At least one parameter is invalid 4 = Acknowledge, command will be performed with completion signaled later by an event 5 = Rejected, Already in Desired Condition 6 = No such object exists 7-63 Reserved	S2F42, F50
HOACK	11	Conveys whether the corresponding handoff activity succeeded (= True) or failed (= False).		S4F31, F33
HOCANCELACK	51	Tells whether the cancel ready message was accepted or rejected.	0 = Cancel Ready Accepted 1 = Atomic Transfer Unknown 2 = Cancel Ready Rejected - Handoff Begun	S4F13
HOCMDNAME	20	Identifier for the handoff command to be executed.		S4F31
HOHALTACK	51	Tells whether the halt command was accepted or rejected.	0 = Halt Accepted 1 = Atomic Transfer Unknown 2-63 Reserved	S4F41

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
IACDS	32, 52	Immediately After Command Codes.	Vector of all command codes which defined command must immediately succeed within the same block.	S7F22
IBCDs	32, 52	Immediately Before Command Codes.	Vector of all command codes which defined command must immediately precede within the same block.	S7F22
IDTYP	10	Id type.	0 = Wafer ID 1 = Wafer Cassette ID 2 = Film Frame ID >2 = Error 3-63 Reserved	S12F1, F3, F4, F5, F7, F9, F11, F13, F14, F15, F16, F17, F18
INPTN	10, 51	A specialized version of PTN indicating the InputPort.		S3F35
JOB ACTION	20	Specifies the action for a ReticleTransferJob.		S3F35
LENGTH	3(), 5()	Length of the service program or process program in bytes.		S2F1; S7F1, F29
LIMITACK	10	Acknowledgment code for variable limit attribute set, 1 byte.	1 = LIMITID does not exist 2 = UPPERDB > LIMITMAX 3 = LOWERDB < LIMITMIN 4 = UPPERDB < LOWERDB 5 = Illegal format specified for UPPERDB or LOWERDB 6 = ASCII value cannot be translated to numeric 7 = Duplicate limit definition for this variable >7 = Other equipment-specific error 8-63 Reserved	S2F46
LIMITID	10	The identifier of a specific limit in the set of limits (as defined by UPPERDB and LOWERDB) for a variable to which the corresponding limit attributes refer, 1 byte.		S2F45, F46, F48
LIMITMAX	11, 20, 3(), 4(), 5()	The maximum allowed value for the limit values of a specific variable. The equipment manufacturer should specify this value, which would typically coincide with the maximum value of the variable being monitored. The format must match that of the referenced variable.		S2F48

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
LIMITMIN	11, 20, 30, 40, 50	The minimum allowed value for the limit values of a specific variable. The equipment manufacturer should specify this value, which would typically coincide with the minimum value of the variable being monitored. The format must match that of the referenced variable.		S2F48
LINKID	54	Used to link a completion message with a request that an operation be performed. LINKID is set to the value of RMOPID in the initial request except for the last completion message to be sent, where it is set to zero.		S6F25; S14F20, F21; S15F22, F30
LLIM	30, 40, 50	Lower limit for numeric value.		S7F22
LOC	10	Machine material location code, 1 byte.		S2F27; S3F2
LOCID	20	The logical identifier of a material location.		S3F29, F31
LOWERDB	11, 20, 30, 40, 50	A variable limit attribute which defines the lower boundary of the deadband of a limit. The value applies to a single limit (LIMITID) for a specified VID. Thus, UPPERDB and LOWERDB as a pair define a limit.		S2F45, F48
LRACK	10	Link Report Acknowledge Code, 1 byte.	0 = Accepted 1 = Denied. Insufficient space 2 = Denied. Invalid format 3 = Denied. At least one CEID link already defined 4 = Denied. At least one CEID does not exist 5 = Denied. At least one RPTID does not exist >5 = Other errors 6-63 Reserved	S2F36
LVACK	10	Variable limit definition acknowledge code, 1 byte. Defines the error with the limit attributes for the referenceVID.	1 = Variable does not exist 2 = Variable has no limits capability 3 = Variable repeated in message 4 = Limit value error as described in LIMITACK 5-63 Reserved	S2F46

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
MAPER	10	Map Error.	0 = ID not found 1 = Invalid Data 2 = Format Error >2 = Invalid error 3-63 Reserved	S12F19
MAPFT	10	Map data format type.	0 = Row format 1 = Array format 2 = Coordinate format >2 = Error 3-63 Reserved	S12F3, F5
MCINDEX	5()	Identifier used to link a handoff command message with its eventual completion message. Corresponding messages carry the same value for this data item.		S4F31, F33
MDACK	10	Map data acknowledge.	0 = Map received 1 = Format error 2 = No ID match 3 = Abort/discard map >3 = Error 4-63 Reserved	S12F8, F10, F12
MDLN	20	Equipment Model Type, 20 bytes max.	Same data as returned by S1, F2	S1F2, F13, F14; S7F22, F23, F26, F31, F39, F43
MEXP	20	Message expected in the form SxxFyy where x is stream and y is function.		S9F13

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
MF	10, 20	Material format code 1 byte by Format 10.	<p>Items with format 10 will be encoded as follows:</p> <p>1 = Quantities in wafers</p> <p>2 = Quantities in cassette</p> <p>3 = Quantities in die or chips</p> <p>4 = Quantities in boats</p> <p>5 = Quantities in ingots</p> <p>6 = Quantities in leadframes</p> <p>7 = Quantities in lots</p> <p>8 = Quantities in magazines</p> <p>9 = Quantities in packages</p> <p>10 = Quantities in plates</p> <p>11 = Quantities in tubes</p> <p>12 = Quantities in waterframes</p> <p>13 = Quantities in carriers</p> <p>14 = Quantities in substrates</p> <p>15-63 Reserved</p> <p>Items with format 20 will be a unit identifier for one of the special SECS generic units, as specified in § 12.</p>	S3F2, F4, F5, F7; S16F11, F13, F15
MHEAD	10	SECS message block header associated with message block in error.		S9F1, F3, F5, F7, F11
MID	10, 20	Material ID.	80 Characters maximum	S2F27; S3F2, F4, F7, F9, F12, F13; S4F1, F3, F5, F7, F9, F11, F13, F15, F17; S7F7, F8, F10, F11, F13, F35, F36; S12F1, F3, F4, F5, F7, F9, F11, F13, F14, F15, F16, F17, F18; S16F11, F13, F15; S18F10, F11
MIDAC	10	Material ID Acknowledge Code, 1 byte.	<p>0 = Accepted</p> <p>1 = Invalid port number</p> <p>2 = Material is not present at identified port</p> <p>&gt;2 = Error</p> <p>3-63 Reserved</p>	S3F14

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
MIDRA	10	Material ID Acknowledge Code, 1 byte.	0 = Acknowledge, MID follows 1 = Acknowledge, will not send MID 2 = Acknowledge, will send MID later in S3F13 3-63 Reserved	S3F12
MLCL	5()	Message length.	Defined by message size in (bytes)	S12F4, F5
MMODE	10	Matrix mode select, 1 byte.	1 = Host source mode Use S7F7 and F8 to define the process program to be used. 2 = Local source mode Use the matrix defined by S7F9, F10, F11 and F13 to define the process program to be used. The equipment will initialize to local source mode at power up. The equipment will default to local source mode from host source mode in the event of loss of communication with the host. 3 = Host immediate mode Use the current process program for all material unless changed by S7F1-F4. The timing of the mode change is equipment-specific.	S7F15
NACDS	32, 52	Not After Command Codes.	Vector of all command codes which defined command may not succeed within the same block.	S7F22
NBCDS	32, 52	Not Before Command Codes.	Vector of all command codes which defined command may not precede within the same block.	S7F22
NULBC	20, 51	Null bin code value.	This value is the bin code value that is used for no die at a location. (For X/Y coordinate format the ASCII value is a value with "n" length. For other map formats, ASCII is a single byte per bin with "n" length per item; thus, the total number of bins is the length, i.e., length "n" = 10 for ASCII format is 10 single byte bin codes.) The format used must be the same as the one used for BINLT and BCEQU. Zero length indicates not used.	S12F1, F3, F4
OBJACK	51	Acknowledge code:	0 = Successful completion of requested data 1 = Error >1 Reserved	S14F2, F4, F6, F8, F10, F12, F14, F16, F18, F26, F28

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
OBJCMD	51	Specifies an action to be performed by an object:	0 = Reserved 1 = Attach to requestor 2 = Detach from requestor (requires authorization token) 3 = Reattach to requestor 4 = Set attributes (requires authorization token) >4 Reserved	
OBJID	20, 5()	Identifier for an object.		S1F19; S14F1, F2, F3, F4
OBJSPEC	20	A text string that has an internal format and that is used to point to a specific object instance. The string is formed out of a sequence of formatted substrings, each specifying an object's type and identifier. The substring format has the following four fields: object type, colon character “:”, object identifier, greater-than symbol “>” where the colon character “:” is used to terminate an object type and the “greater than” symbol “>” is used to terminate an identifier field. The object type field may be omitted where it may be otherwise determined. The final “>” is optional.		S2F49; S13F11, F13, F15; S14F1, F3, F5, F7, F9, F10, F11, F13, F15, F16, F17, F19, F25, F27; S15F43, F47
OBJTOKEN	54	Token used for authorization.		S14F14, F15; S15F37, F39, F41, F43
OBJTYPE	20, 5()	Identifier for a group or class of objects. All objects of the same type must have the same set of attributes available.		S1F19; S14F1, F3, F6, F7, F8, F25, F26, F27
OFLACK	10	Acknowledge code for OFF-LINE request.	0 = OFF-LINE Acknowledge 1-63 Reserved	S1F16
ONLACK	10	Acknowledge code for ON-LINE request.	0 = ON-LINE Accepted 1 = ON-LINE Not Allowed 2 = Equipment Already ON-LINE 3-63 Reserved	S1F18
OPID	5()	Operation ID. A unique integer generated by the requestor of an operation, used where multiple completion confirmations may occur.		S6F25; S14F19, F21; S15F21, F29, F30, F37, F41, F44, F46

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
ORLOC	10	Origin Location.	implicit value of (0,0) 0 = Center die of wafer (Determined by:  $\frac{\text{row or column count} + 1}{2}$ truncated) 1 = Upper right 2 = Upper left 3 = Lower left 4 = Lower right >4 = Error 5-63 Reserved Zero length indicates not available	S12F1, F3, F4
OUTPTN	10, 51	A specialized version of PTN indicating the OutPutPort.		S3F35
PARAMNAME	20	The name of a parameter in a request.		S3F21, F23
PARAMVAL	1, 10, 11, 20, 3(), 4(), 5()	The value of the parameter named in PARAMNAME. Values that are lists are restricted to lists of single items of the same format type.		S3F21, F23
PDEATTRIBUTE	51	Selection from available PDE attributes whose values could be reported.	0 RESERVED 1 = name 2 = gid 3 = groupName 4 = description 5 = type 6 = executable 7 = maxAntecedents 8 = createDate 9 = createNode 10 = author 11 = userInfo 12 = supplierInfo 13 = checksum 14-255 Reserved	S19F1, F2
PDEATTRIBUTENAME	51	Selection from available PDE attributes that can be used to filter the PDE directory report.	0 = uid 1 = name 2 = gid 3 = groupName 4 = description 5 = type 6 = executable 7 RESERVED 8 = createDate 9 = createNode 10 = author 11 = userInfo 12 = supplierInfo 13-255 Reserved	S19F1



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
PDEATTRIBUTEVALUE	00, 11, 20, 21, 51	Contains the value of the corresponding PDEATTRIBUTE in the appropriate format.	Formats: 00 used for list of strings (20 or 21) for userInfo and supplierInfo 11 used for executable 20 used for uid and gid and also for createDate and for checksum (hex string) 20 or 21 used for name, groupName, description, type, createNode, author 51 used for maxAntecedents	S19F1, F2
PDEREF	20	Contains the unique identifier of a PDE (uid) or of a PDE group (gid).	36 character string in groupings of 8, 4, 4, 4, and 12 separated by hyphens "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"	S19F15, F16, F17
PDFLT	11, 20, 3(), 4(), 5()	Parameter Default Value.	Specifies default value and data type of parameter. If no defaults are provided, item data length will be zero. For numeric or Boolean data, the default item may be a multi-varied vector if the parameter itself can be a vector. If RQPAR is false: Position of data in a default item is significant. When obtaining a default value to be used in the Nth position of a vector parameter, the value is obtained from the Nth position of the default item. If the default item has L entries, no default value will be provided for the L+1,...,PMAX parameter entries. If RQPAR is true: The length of the default vector (L) specifies the minimum number of entries which must be entered for the parameter. If > PMAX, only PMAX entries are required.	S7F22
PFCD	10	Predefined form code, 1 byte.		S6F9
PGRPACTION	20	The action to be performed on a port group.		S3F23

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
PMAX	3(), 5()	Parameter Count Maximum.	Maximum amount of data to be accepted by the host for this parameter. When a conflict arises between value of PMAX and length of PDFLT, PMAX takes precedence. For numeric and Boolean parameters: PMAX < 0 invalid PMAX = 0 specifies there is no upper bound PMAX = 1 specified a single value is expected PMAX > 1 specifies a vector of values is expected with a maximum of PMAX entries For string parameters: PMAX < 0 invalid PMAX = 0 specifies there is no upper bound PMAX > 0 maximum length of parameter string	S7F22
PNAME	20	Parameter Name ≤16 characters.	Text string identifying the parameter value expected by its parent process command.	S7F22
PORTACTION	20	The action to be performed on a port.		S3F25
PORTGRPNAME	20	The identifier of a group of ports.		S3F21,F23
PPARM	11, 20, 3(), 4(), 5()	Process Parameter.	Numeric or Boolean SECS data item, single or multiple value, or text string which provides information required to complete the process command to which the parameter refers.	S7F23, F26, F31, F39, F43
PPBODY	10, 20, 3(), 5()	Process program body.	The process program describes to the equipment, in its own language, the actions to be taken in processing the material it receives.	S7F3, F6, F36, F37, F41
PPGNT	10	Process program grant status, 1 byte.	0 = OK 1 = Already have 2 = No space 3 = Invalid PPID 4 = Busy, try later 5 = Will not accept >5 = Other error 6-63 Reserved	S7F2, F30

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
PPID	10, 20	Process program ID.	Limited to a maximum of 120 bytes. The format used in the PPID will be host-dependent. For internal use of the equipment, the PPID can be treated as a unique binary pattern. If the local equipment is not prepared to display the transmitted code, the display should be in hexadecimal form.	S2F27; S7F1, F3, F5, F6, F8, F10, F11, F13, F17, F20, F23, F25, F26, F27, F31, F33, F34, F36, F53; S9F13
PRAXI	10	Process axis.	0 = Rows (X-axis), top, increasing 1 = Rows (X-axis), top, decreasing 2 = Rows (X-axis), bottom, increasing 3 = Rows (X-axis), bottom, decreasing 4 = Columns (Y-axis), left, increasing 5 = Columns (Y-axis), left, decreasing 6 = Columns (Y-axis), right, increasing 7 = Columns (Y-axis), right, decreasing >7 = Error 8-63 Reserved	S12F1, F3
PRCMDNAME	20	Commands sent to a process job:	"START" "STOP" "PAUSE" "RESUME" "ABORT" "CANCEL"	S16F5
PRDCT	5()	Process Die Count.	Number of die to be processed or number of die, which have been processed. (Zero length indicates not used.)	S12F1, F4
PREVENTID	5()	Processing related event identification:	1 = Waiting for material 2 = Job state change	S16F9
PRJOBID	20	Text string which uniquely identifies a process job.		S16F4, F5, F6, F7, F9, F11, F12, F13, F14, F15, F16, F17, F20, F23, F25
PRJOBMILESTONE	5()	Notification of Processing status shall have one of the following values:	1 = Job Setup 2 = Job Processing 3 = Job Processing Complete 4 = Job Complete 5 = Job Waiting for Start	S16F7
PRJOBSPACE	52	The number of process jobs that can be created.		S16F22

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
PRMTRLORDER	51	Defines the order by which material in the process jobs material list will be processed. Possible values are assigned as follows:	1 = ARRIVAL - process whichever material first arrives 2 = OPTIMIZE - process in an order that maximizes throughput 3 = LIST - follow the order in the list	S16F29
PRPAUSEEVENT	00	The list of event identifiers, which may be sent as an attribute value to a process job. When a process job encounters one of these events it will pause, until it receives the PRJobCommand RESUME.		S16F11, F13, F15
PRPROCESSSTART	11	Indicates that the process resource start processing immediately when ready:	TRUE = Automatic Start FALSE = Manual Start	S16F11, F13, F15, F25
PRRECIPEMETHOD	51	Indicates the recipe specification type, whether tuning is applied and which method is used:	1 - Recipe only 2 - Recipe with variable tuning	S16F11, F13, F15
PRSTATE	51	Enumerated value, 1 byte.		S16F20
PTN	10, 51	Material Port number, 1 byte.		S3F17, F21, F25; S4F1, F3, F5, F7, F9, F11, F13, F15, F17; S16F13, F17, F21
QUA	10	Quantity in format, 1 byte.		S3F2, F4, F5, F7
RAC	31, 51	Reset acknowledge, 1 byte.	0 = Reset to be done 1 = Reset denied >1 = Other errors 2-63 Reserved	S2F20
RCMD	20, 31, 51	Remote command code or string.		S2F21, F41, F49
RCPATTRDATA	0, 10, 11, 20, 3(), 4(), 5()	The contents (value) of a recipe attribute.		S6F25; S15F13, F15, F18, F27, F28, F30, F32, F49, F51
RCPATTRID	20	The name (identifier) of a non-identifier recipe attribute.		S6F25; S15F13, F15, F18, F27, F28, F30, F32, F49, F51
RCPBODY	10, 20, 3(), 5()	Recipe body.		S15F13, F15, F18, F27, F32, F49, F51
RCPCCLASS	20	Recipe class.		S15F11

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RCPCMD	51	Indicates an action to be performed on a recipe.	5 = Delete 8 = Unprotect 9 = Protect 10 = Verify 11 = Link 12 = Unlink 13 = Certify 14 = De-certify 15 = Download 16 = Upload 0-4, 6-7, 17-63 Reserved	S15F21, F22
RCPDEL	51		0 = Delete 1 = Deselect >1 Reserved	S15F35
RCPDESLTH	5()	The length in bytes of a recipe section.		S15F24
RCPDESCNM	20	Identifies a type of descriptor of a recipe: "ASDesc", "BodyDesc", "GenDesc."		S15F24
RCPDESTIME	20	The timestamp of a recipe section, in the format "YYYYMMDDhhmmsscc."		S15F24
RCPID	20	Recipe identifier. Formatted text conforming to the requirements of OBJSPEC.		S15F21, F23, F28, F29, F30, F33, F35, F37, F41, F44, F53
RCPNAME	20	Recipe name.		S15F11
RCPNEWID	20	The new recipe identifier assigned as the result of a copy or rename operation.		S15F19, F41, F44
RCPOWCODE	11	Indicates whether any pre-existing recipe is to be overwritten (= TRUE) or not (= FALSE) on download.		S15F27, F49
RCPPARNM	20	The name of a recipe variable parameter. Maximum length of 256 characters.		S15F25, F33; S16F3, F11, F13, F15, F23
RCPPARRULE	20	The restrictions applied to a recipe variable parameter setting. Maximum length of 80 characters.		S15F25
RCPPARVAL	0, 10, 11, 20, 3(), 4(), 5()	The initial setting assigned to a recipe variable parameter. Text form restricted to maximum of 80 characters.		S15F25, F33; S16F3, F11, F13, F15, F23
RCPRENAME	11	Indicates whether a recipe is to be renamed (= TRUE) or copied (= FALSE).		S15F19

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RCPSECCODE	10	Indicates the sections of a recipe requested for transfer or being transferred:	1 = Generic attributes only 3 = Generic attributes and body 4 = All agent-specific datasets (zero or more) 7 = Generic attributes, body, and all agent-specific datasets 8 = Single agent-specific dataset (zero or one) 11 = Generic attributes, body, and single agent-specific datasets All other values reserved	S15F15, F16, F17
RCPSECNM	20	Recipe section name: "Generic", "Body", or "ASDS."		S15F15, F18
RCPSPEC	20	Recipe specifier. The object specifier of a recipe.		S15F1, F9, F13, F15, F17, F19, F27, F28, F31, F32, F45, F49, F51, F53; S16F11, F13, F15, F23
RCPSTAT	51	The status of a managed recipe.	0 = Does not exist 8 = Unprotected 9 = Protected	S15F10
RCPUPDT	11	Indicates if an existing recipe is to be updated (= True) or a new recipe is to be created (= False).		S15F13
RCPVERS	20	Recipe version.		S15F10, F12
READLN	3(), 5()	Maximum length to read.		S13F5
RECLEN	3(), 5()	Maximum length of a Discrete record.		S13F4
REFP	3()	Reference Point.		S12F1, F4
REPGSZ	20, 3(), 5()	Reporting group size.		S2F23; S17F5
RESC	31, 51	Resolution code for numeric data.	1 = Absolute. Value may be specified to nearest increment of RESV 2 = Significant Digits. Value may be specified with no more significant digits than RESV allows	S7F22
RESOLUTION	20	Contains the unique identifier of a PDE (uid).	36 character string in groupings of 8, 4, 4, 4, and 12 separated by hyphens "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"	S19F15, F16, F17

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RESPDESTAT	51	Status response for the Resolve PDE request. If more than one of these conditions applies, the first value on the list that applies should be returned.	0 = OK - Indicates successful completion; no problems were encountered. 1 = InvalidInputMap - The supplied <i>inputMap</i> was improperly formatted. This response shall be reported only for the <i>targetPDE</i> . 2 = MapPDENotFound - A PDE reference was resolved in the <i>inputMap</i> , but not found in the local store. 3 = ResolveDenied - The equipment was not allowed to resolve this <i>PDE</i> reference. 4 = ReferencedPDENotFound - No <i>PDE</i> could be found in the local store to satisfy one or more <i>PDE</i> references within this <i>PDE</i> . 5 = PDENotFound - The specified <i>PDE</i> was not found in the local store. 6 = DateTimeCompareFailed - The <i>createDate</i> fields of the <i>PDE</i> 's within a group could not be compared successfully. 7-254 Reserved 255 = Other - A problem occurred that is not described by the other enumerated values.	S19F16
RESPEC	20	Object specifier for the recipe executor.		S15F29, F33, F35
RESV	3(), 4(), 5()	Resolution value for numeric data.	If RESC=1, then RESV contains smallest increment allowed for parameter	S7F22
RETICLEID	20	The object identifier for a reticle. Conforms to OBJSPEC.		S3F35
RETPLACEINSTR	51	Instructions to indicate which pod slots will have reticles placed. Possible values for Reticle-PlacementInstruction are:	0 = PLACE 1 = PASS BY 2 = CURRENTLY OCCUPIED	S3F35
REREMOVEINSTR	51	Instructions to indicate which pod slots will have reticles removed.	0 = REMOVE 1 = PASS BY	S3F35
RIC	31, 51	Reset code, 1 byte.	0 = Not used 1 = Power up reset >1 Other reset conditions 2-63 Reserved	S2F19

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RMACK	51	Conveys whether a requested action was successfully completed, denied, completed with errors, or will be completed with notification to the requestor.	0 = Successful completion 1 = Cannot perform action 2 = Completed with errors 3 = Action will be completed and notification sent 4 = No request for this action exists	S15F4, F6, F8, F10, F12, F14, F16, F18, F20, F22, F24, F25, F26, F28, F30, F32, F34, F36, F38, F40, F42, F44, F48, F53
RMCHGSTAT	5()	Indicates the change that occurred for an object.	0 = No change 1 = Created 2 = Updated 3 = Stored (new) 4 = Replaced 5 = Deleted 6 = Copied (new object) 7 = Renamed 8 = Unprotected 9 = Protected 10 = Verified 11 = Linked 12 = Unlinked 13 = Certified 14 = De-certified 15 = Selected 16 = Deselected	S15F25
RMCHGTYPE	5()	Indicates the type of change for a recipe.	0 = No change 1 = Create 2 = Update 5 = Delete 6 = Copy (new object) 7 = Rename 8 = Unprotect 9 = Product 10 = Verify 11 = Link 12 = Unlink 13 = Certify 14 = De-certify 15 = Change generic attribute 16 = Change agent-specific attribute 17 = Change both generic and agent-specific attributes	S15F37, F41, F44, F46, F47, F48
RMDATASIZE	5()	The maximum total length, in bytes, of a multi-block message, used by the receiver to determine if the anticipated message exceeds the receiver's capacity.		S15F1



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RMGRNT	10	Grant code, used to grant or deny a request. 1 byte.	0 = Permission granted 1 = Cannot accept now, try again 2 = No space 3 = Request is on hold 4-64 Reserved	S15F2, F37, F46
RMNEWS	20	New name (identifier) assigned to a recipe namespace.		S15F5
RMNSCMD	51	Action to be performed on a recipe namespace.	1 = Created 5 = Deleted 0, 2-4, 6-63 Reserved	S15F3
RMNSSPEC	20	The object specifier of a recipe namespace.		S15F3, F5, F7, F11, F21, F23, F25, F47
RMRECSPEC	20	The object specifier of a distributed recipe namespace recorder.		S15F39, F41, F43
RMREQUESTOR	11	Set to TRUE if initiator of change request was an attached segment. Set to FALSE otherwise.		S15F41, F44, F46
RMSEGSPEC	20	The object specifier of a distributed recipe namespace segment.		S15F37, F39, F41, F44, F46, F47
RMSPACE	5()	The amount of storage available for at least one recipe in a recipe namespace, in bytes.		S15F8
ROWCT	5()	Row count in die increments.		S12F1, F4
RPMACK	51	Reticle Pod management service acknowledge code. 1 byte.	0 = Acknowledge, service has been performed 1 = Service does not exist 2 = Can not perform now 3 = At least parameter does not exist 4 = Acknowledge, request will be performed with completion signaled later by an event 5 = Service is not completed or prohibited 6 = No such object exists 7-63 Reserved	
RPMDESTLOC	20	The LocationID towards which a reticle must be moved. Conforms to OBJID.		S14F19
RPMSOURLOC		The LocationID of the location from which to pick-up a reticle for moving it to another location. Conforms to OBJID.		S14F19

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RPSEL	51	Reference Point Select.	Number of reference point from 0-n.	S12F1, F4
RPTID	20, 30, 50	Report ID.		S2F33, F35; S6F11, F13, F16, F11, F19, F21, F22; S17F1, F2, F3, F4, F5, F9, F11, F12
RPTOC	11	A Trace Object attribute for a flag which, if set TRUE, causes only variables which have changed during the sample period to be included in a report.		S14F1, F2, F3, F4; S17F5
RQCMD	11	Required Command.	True = Command must be specified False = Command is optional	S7F22
RQPAR	11	Required Parameter.	True = Parameter must be specified False = Parameter is optional	S7F22
RRACK	10	Request to Receive Acknowledge code, 1 byte.	0 = Acknowledge, OK (note that 'OK' differs from 'ready') 1 = Invalid port number 2 = Requested material is not at identified port 3 = Busy. Try again 4 = Sender does not have permission to perform this operation 5-63 Reserved	S4F18
RSACK	10	Ready to Send Acknowledge code, 1 byte.	0 = Acknowledge, OK (note that 'OK' differs from 'ready') 1 = Invalid port number 2 = Port is already occupied 3 = Busy, unable to move material at this time. Try again 4 = Receiver does not have permission to perform this operation 5-63 Reserved	S4F2
RSDA	10	Request Spool Data Acknowledge.	0 = OK 1 = Denied, busy try later 2 = Denied, spooled data does not exist 3-63 Reserved	S6F24
RSDC	51	Request Spool Data Code.	0 = Transmit Spooled Messages 1 = Purge Spooled Messages 2-63 Reserved	S6F23

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
RSINF	3()	Starting location for row or column. This item consists of 3 values (x,y,direction). If direction value is negative, it equals decreasing direction. If the value is positive, it equals increasing direction. Direction must be a non-zero value.		S12F7, F14
RSPACK	10	Reset Spooling Acknowledge.	0 = Acknowledge, spooling setup accepted 1 = Spooling setup rejected 2-63 Reserved	S2F44
RTSRPSTAT	51	Status response for the Ready To Send request.	0 = OK - Indicates permission to send is granted 1 = NoResources - Indicates there are not sufficient resources currently available to service this request (try again later) 2 = TransferContainerTooLarge - Indicates that the <i>TransferContainer</i> is too large to be processed successfully. 3 = NoStorageSpace - Indicates that the <i>PDE's</i> to be transferred will not fit in available storage space. 4-254 Reserved 255 = Other - A problem occurred that is not described by the other enumerated values.	S19F10
RTYPE	3(), 5()	Type of record.	0 = Stream 1 = Discrete 2-63 Reserved	S13F4
SDACK	10	Map set-up data acknowledge.	0 = Received data >1 = Error 1-63 Reserved	S12F2
SDBIN	10	Send bin information flag.	0 = Sent bin information 1 = Don't send bin information >1 = Error 2-63 Reserved	S12F17

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
SENDRESULT	51	Reports overall success or failure of the sendPDE() request.	0 = OK – All PDE's were transferred successfully. 1 = SendRsPError – A valid TransferContainer was successfully transferred, but one or more PDE's were found to have errors. See sendRspInfo. 2 = TCManifestError – Manifest is missing, corrupt, misformatted or contains extraneous information. 3 = TCUnexpectedContent – TransferContainer contains files not listed in Manifest. 4 = TCMissingFiles – PDE's listed in Manifest are not found in TransferContainer. 5 = TCFormatError – TransferContainer is misformatted or corrupt.	S19F12, F13
SENDRSPSTAT	51	Status response for the Send PDE request.	0 = OK – Indicates PDE was successfully received. 1 = NoResources – Indicates there are not sufficient resources available to receive this PDE (e.g., disk space). 2 = NoExecutionTarget – No equipment component matches the specified <i>ExecutionTarget</i> . 3 = PDElocked – The PDE already exists and cannot be replaced at this time. 4 = VerificationFailed – The PDE failed verification. See VERIFYRSPSTAT for failure reason. 5 = TransferNotAllowed – Transfer of this PDE or type of PDE is prohibited. 6-254 Reserved 255 = Other – A problem occurred that is not described by the other enumerated values.	S19F12, F13
SEQNUM	3(), 5()	Command Number.	Value which identifies a unique process program command by its position in the list of commands relative to the first. For the first command of the process program, SEQNUM is 1.	S7F27
SFCD	10	Status form code, 1 byte.		S1F5

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
SHEAD	10	Stored header related to the transaction timer.		S9F9
SLOTID	51	Used to reference material by slot (a position that holds material/substrates) in a carrier. This item may be implemented as an array in some messages.		S16F11, F13, F15
SMPLN	3(), 5()	Sample number		S6F1
SOFTREV	20	Software revision code 20 bytes maximum.		S1F2, F13, F14; S7F22, F23, F26, F31, F39, F43
SPAACK	10	Equipment acknowledgement code, 1 byte.	0 = Everything correct 1 = Invalid data >1 = Equipment-specific error 2-63 Reserved	S2F4
SPD	10	Service program data.		S2F3, F6
SPID	20	Service program ID, 6 characters.		S2F1, F5, F7, F9, F12; S9F13
SPNAME	20	Service parameter name defined in specific standard. If service parameter is defined as an object attribute, this is completely the same as ATTRID except format restrictions above.		S14F19, F20, F21, F28
SPR	Device Dependent	Service program results.	Device dependent	S2F10
SPVAL	0, 10, 11, 20, 21, 3(), 4(), 5()	Service parameter value, corresponding to SPNAME. If service parameter is defined as an object attribute, this is completely the same as ATTRDATA except format restrictions for the attribute.		S14F19, F20, F21
SSACK	20	Indicates the success or failure of a requested action. Two characters.		S18F2, F4, F6, F8, F10, F12, F14
SSCMD	20	Indicates an action to be performed by the subsystem.		S18F13
STATUS	20	Provides status information for a subsystem component. Used in the data item STATUSLIST.		See STATUSLIST
STATUSLIST	0	A list of STATUS data sent in a fixed order. STATUSLIST has the following form:	L, S 1. <STATUS <sub>1</sub> > . . S. <STATUS <sub>n</sub> >	S18F4, F8, F10, F12, F14, F16

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
STATUSTXT	20	Text string describing the corresponding status response. Maximum length of 80 characters.		S19F2, F4, F6, F8, F10, F12, F13, F16, F18
STEMP	20	String template. ASCII text string acceptable to equipment as a parameter value. A data string matches a template string if the data string is at least as long as the template and each character of the data string matches the corresponding character of the template. A null list indicates all user data is acceptable to the machine.		S7F22
STIME	20	Sample time, 12, 16 bytes, or Extended format as specified by the TimeFormat equipment constant value setting.	Where:  12-byte format YYMMDDhhmmss  16-byte format YYYYMMDDhhmmsscc  Extended (max 32 byte) format YYYY-MM-DDThh:mm:ss.STZD (see SEMI E148)  See TimeFormat equipment constant variable item in table 4 for additional detail on the formats.	S6F1
STRACK	10	Spool Stream Acknowledge.	1 = Spooling not allowed for stream (i.e., Stream 1) 2 = Stream unknown 3 = Unknown function specified for this stream 4 = Secondary function specified for this stream	S2F44
STRID	51	Stream Identification.		S2F43, F44
STRP	3()	Starting position in die coordinate position. Must be in (X,Y) order.		S12F9, F16
SV	0, 10, 11, 20, 21, 3(), 4(), 5()	Status variable value.		S1F4; S6F1

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
SVCACK	10	Service acceptance acknowledge code, 1 byte.	0 = Acknowledge, service has been performed 1 = Service does not exist 2 = Cannot perform now 3 = At least parameter is invalid 4 = Acknowledge, service will be performed with completion notified later with parameters for response 5 = Service is not completed or prohibited 6 = No such object exists 7-63 Reserved	S14F20
SVCNAME	20	Service name provided on specified object asking by the host.		S14F19, F26, F27, F28
SVID	20, 3(), 5()	Status variable ID.	Status variables may include any parameter that can be sampled in time such as temperature or quantity of a consumable.	S1F3, F11, F12; S2F23
SVNAME	20	Status Variable Name.		S1F12
TARGETID	20	Identifies where a request for action or data is to be applied. If text, conforms to OBJSPEC.		S18F1, F3, F5, F7, F9, F11, F13
TARGETPDE	20	Contains the unique identifier (uid) of the PDE that is the starting point for the verification process.	36 character string in groupings of 8, 4, 4, 4, and 12 separated by hyphens "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"	S19F15
TARGETSPEC	20	Object specifier of target object.		S14F17; S15F43
TBLACK	51	Indicates success or failure.	0 = Success 1 = Failure	S13F14, F16
TBLCMD	51	Provides information about the table or parts of the table being transferred or requested. Enumerated:	0 = Complete Table 1 = New rows (add) 2 = New columns (append) 3 = Replace existing rows 4 = Replace existing columns	S13F13, F15
TBLELT	0, 10, 11, 20, 21, 3(), 4(), 5()	Table element. The first table element in a row is used to identify the row.		S13F13, F15, F16
TBLID	20	Table identifier. Text conforming to the requirements of OBJSPEC.		S13F13, F15, F16
TBLTYP	20	A reserved text string to denote the format and application of the table. Text conforming to the requirements of OBJSPEC.		S13F13, F15, F16

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
TCID	20	TCID is the identifier of the TransferContainer.	36 character string in groupings of 8, 4, 4, 4, and 12 separated by hyphens "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"	S19F6, F8, F9, F10, F11
TEXT	10, 20, 22, 3(), 5()	A single line of characters.		S10F1, F3, F5, F9
TIAACK	10	Equipment acknowledgement code, 1 byte.	0 = Everything correct 1 = Too many SVIDs 2 = No more traces allowed 3 = Invalid period 4 = Unknown SVID specified 5 = Invalid REPGSZ 6-63 Reserved >63 Equipment specified error	S2F24
TIACK	10	Time Acknowledge Code, 1 byte.	0 = OK 1 = Error, not done 2-63 Reserved	S2F32
TID	10	Terminal number, 1 byte.	0 = Single or main terminal >0 = Additional terminals at the same equipment	S10F1, F3, F5, F7
TIME	20	Time of day, 12, 16 bytes, or Extended format as specified by the TimeFormat equipment constant value setting.	Where:  12-byte format YYMMDDhhmmss  16-byte format YYYYMMDDhhmmsscc  Extended (max 32 byte) format YYYY-MM-DDThh:mm:ss.sTZD (see SEMI E148)  See TimeFormat equipment constant variable item in Table 4 for additional detail on the formats.	S2F18, F31
TIMESTAMP	20	Timestamp in 12, 16 bytes, or Extended format indicating the time of an event, which encodes time as specified by the TimeFormat equipment constant value setting.	Where:  12-byte format YYMMDDhhmmss  16-byte format YYYYMMDDhhmmsscc  Extended (max 32 byte) format YYYY-MM-DDThh:mm:ss.sTZD (see SEMI E148)  See TimeFormat equipment constant variable item in Table 4 for additional detail on the formats.	S5F9, F11, F15; S15F41, F44; S16F5, F7, F9



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
TOTSMP	20, 3(), 5()	Total samples to be made.		S2F23; S17F5
TRACK	11	Tells whether the related transfer activity was successful (= True) or unsuccessful (= False).		S4F20, F22, F23
TRANSFERSIZE	50	Size, in bytes, of the TransferContainer proposed for transfer.		S19F9
TRATOMICID	5()	Equipment assigned identifier for an atomic transfer.		S4F20
TRAUTOD	11	A Trace Object attribute for a control flag which, if set TRUE, causes the Trace Object to delete itself when it has completed a report.		S14F1, F2, F3, F4; S17F5
TRAUTOSTART	11	For each atomic transfer, this data item tells the equipment if it should automatically start the handoff when ready (= TRUE) or await the host's "StartHandoff" command (= FALSE) following setup. This data item only affects the primary transfer partner.		S4F19
TRCMDNAME	20	Identifier of the transfer job-related command to be executed. Possible values:	"CANCEL" "PAUSE" "RESUME" "ABORT" "STOP" "STARHANDOFF" (requires a TRATOMICID as a parameter)	S4F21
TRDIR	51	Direction of handoff.	1 = Send material 2 = Receive material 0, 3-63 Reserved	S4F19, F29
TRID	20, 3(), 5()	Trace request ID.		S2F23; S6F1, F27, F28; S17F5, F6, F7, F8, F13, F14
TRJOBID	10	Equipment assigned identifier for the transfer job.		S4F20, F21
TRJOBMS	51	Milestone for a transfer job (e.g., started or complete).	1 = Transfer Job Started 2 = Transfer Job Complete 0, 3-63 Reserved	S4F23
TRJOBNAME	20	Host assigned identifier for the transfer job. Limited to a maximum of 80 characters.		S4F19, F23

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
TRLINK	5()	Common identifier for the atomic transfer used by the transfer partners to confirm that they are working on the same host-defined task.		S4F19, F29, F31, F33, F39
TRLOCATION	5()	Identifier of the material location involved with the transfer. For one transfer partner, this will represent the designated source location for the material to be sent. For the other transfer partner, it will represent the designated destination location for the material to be received.		S4F19, F29
TROBJNAME	20	Identifier for the material (transfer object) to be transferred.		S4F19, F23, F29
TROBJTYPE	5()	Type of object to be transferred.		S4F19, F29
TRPORT	5()	Identifier of the equipment port to be used for the handoff.		S4F19, F29
TRPTNR	20	Name of the equipment which will serve as the other transfer partner for this atomic transfer. This corresponds to EQNAME.		S4F19, F29
TRPTPORT	5()	Identifier of the transfer partner's port to be used for the transfer.		S4F19, F29
TRRCP	20	Name of the transfer recipe for this handoff. Limited to a maximum of 80 characters.		S4F19
TRROLE	51	Tells whether the equipment is to be the primary or secondary transfer partner.	1 = Primary Transfer Partner 2 = Secondary Transfer Partner	S4F19, F29
TRSPER	4()	A Trace Object attribute which holds the value for sampling interval time.		S14F1, F2, F3, F4; S17F5
TRTYPE	51	Tells whether the equipment is to be an active or passive participant in the transfer.	1 = Active 2 = Passive	S4F19, F23, F29
TSIP	10	Transfer status of input port, 1 byte.	1 = Idle state 2 = Prep state 3 = Track on state 4 = Stuck in Receiver state 5-63 Reserved	S1F10

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
TSOP	10	Transfer status of output port, 1 byte.	1 = Idle state 2 = Prep state 3 = Track on state 4 = Stuck in Sender state 5 = Completion state 6-63 Reserved	S1F10
TTC	3(), 5()	Time to completion.		S3F4
UID	20	Contains a unique identifier for a PDE.	36 character string in groupings of 8, 4, 4, 4, and 12 separated by hyphens "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"	S19F2, F3, F4, F5, F6, F7, F8, F12, F13, F18
ULIM	3(), 4(), 5()	Upper limit for numeric value.		S7F22
UNFLEN	3(), 5()	Unformatted Process Program Length.	If greater than zero, indicates PPID is available as an unformatted process program and its length in bytes. If zero, the PPID is not available as an unformatted process program. Negative values are invalid.	S7F34
UNITS	20	Units Identifier.	As allowed by SEMI E5 § 12.	S1F12; S2F30, F48; S7F22
UPPERDB	11, 20, 3(), 4(), 5()	A variable limit attribute which defines the upper boundary of the deadband of a limit. The value applies to a single limit (LIMITID) for a specified VID. Thus, UPPERDB and LOWERDB as a pair define a limit.		S2F45, F48
V	0, 10, 11, 20, 21, 3(), 4(), 5()	Variable data.		S6F11, F13, F16, F20, F22
VERIFYDEPTH	51	Selects whether to check only the target PDE or all associated PDEs within a multi-part recipe.	0 = Single - Only the specified PDE is checked 1 = All - Verify the specified PDE, plus all PDEs it references directly or indirectly. 2-255 Reserved	S19F17

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
VERIFYRSPSTAT	51	Verification result.	<p>0 = OK - Indicates successful completion; no problems were encountered.</p> <p>1 = InvalidInputMap - The supplied <i>inputMap</i> was improperly formatted. This response shall be reported only for the <i>targetPDE</i>.</p> <p>2 = MapPDENotFound - A <i>PDE</i> reference was resolved in the <i>inputMap</i>, but not found in the local store.</p> <p>3 = ResolveDenied - The equipment was not allowed to resolve this <i>PDE</i> reference.</p> <p>4 = ReferencedPDENotFound - No <i>PDE</i> could be found in the local store to satisfy one or more <i>PDE</i> references within this <i>PDE</i> (only for VERIFYDEPTH = 1).</p> <p>5 = DateTimeCompareFailed - The <i>createDate</i> fields of the <i>PDE</i>'s within a group could not be compared successfully.</p> <p>6 = PDENotFound - The specified <i>PDE</i> was not found in the local store.</p> <p>7 = ChecksumFail - The computed checksum of the <i>PDEheader</i> or of the <i>PDEbody</i> did not match the values in the <i>PDE</i>.</p> <p>8 = SyntaxError - The syntax of the <i>PDE</i> or its body was found to be incorrect.</p> <p>9 = ContentError - The content of the <i>PDE</i> did not match allowed values or did not follow all content rules.</p> <p>10 = NoExecutionTarget - No equipment component matches the specified <i>ExecutionTarget</i>.</p> <p>11 = OutputParameterError - The parameter values have not been correctly supplied to all referenced <i>PDE</i>'s.</p> <p>12 = NoVerification - The verification process was not attempted (only for <i>sendPDE</i>).</p> <p>13-254 Reserved</p> <p>255 = Other - A problem occurred that is not described by the other enumerated values.</p>	S19F12, F13, F18

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Where Used</i>
VERIFYSUCCESS	11	Boolean.	True if no errors were encountered during verification.	S19F18
VERIFYTYPE	51	Choice of the type of verification to perform.	0 = Checksum - Calculates the checksum values of the PDE and compares with the embedded value 1 = Validity - Checks the content, syntax, and checksum, to determine if this recipe can be executed on this equipment. 2-255 Reserved	S19F17
VID	20, 3(), 5()	Variable ID.		S2F33, F45, F46, F47, F48; S6F13, F18, F22; S17F1
VLAACK	10	Variable Limit Attribute Acknowledge Code, 1 byte.	0 = Acknowledge, command will be performed 1 = Limit attribute definition error 2 = Cannot perform now >2 = Other equipment-specific error 3-63 Reserved	S2F46
XDIES	4(), 5()	X-axis die size (index).		S12F1, F4
XYPOS	3()	X and Y Coordinate Position. Must be in (X,Y) order.		S12F11, F18
YDIES	4(), 5()	Y-axis die size (index).		S12F1, F4

9.7 *Variable Item Dictionary* — This section defines variable data items which are available to the Host for data collection purposes.

- *Name* — A unique mnemonic name for this variable data item. This name is provided for reference only.
- *Class* — The data type classification (SV, ECV, or DVVAL) of the item. Status values (SVs) always contain valid information, while data values (DVVALs) may only be valid upon the occurrence of a particular event. All equipment constants (ECVs) are settable by the Host.
- *Format* — The allowable item format codes which can be used for this variable data item, as in Data Item Dictionary.
- *Description* — A description of the variable data item, with the meanings of specific values. Also, specify validity for item of class DVVAL.

**Table 4 Variable Item Dictionary**

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
AlarmID	3(), 5()	This variable is valid only upon the setting or clearing of an alarm condition and contains the current alarm identification (ALID), regardless of whether that alarm is enabled for reporting.		DVVAL
AlarmsEnabled	0	Contains the list of alarms (ALIDs) enabled for reporting (via Stream 5).	Structure: L,n    n = # of alarms enabled 1.<ALID <sub>1</sub> > . . n.<ALID <sub>n</sub> >	SV
AlarmsSet	0	Contents of this variable is a list of alarms (ALIDs) currently in the UNSAFE (alarm set) state, regardless of whether the alarms are enabled for reporting.	Structure: L,n    n = # of alarms set 1.<ALID <sub>1</sub> > . . n.<ALID <sub>n</sub> >	SV
ARAMSAccumReset	20	The timestamp of when the set of accumulators EngTime, InterruptionCtr, PrdTime, NSTime, SbyTime, SDTime, and UDTIME were reset to zero. Uses format defined for CLOCK.		SV
ARAMSInfo	20	Text field set by the equipment to provide additional information concerning an ARAMS state change.		SV
ARAMSState	20	The ARAMS code corresponding to the current state/substate. Four characters.		SV
ARAMSText	20	Text describing the ARAMSState. 3–80 characters.		SV
ARAMSTimeStamp	20	The timestamp of the last ARAM state change. This is a format requirement only and does not imply precision or accuracy. Uses format defined for CLOCK.		SV
CLOCK	20	Value of internal clock in 12, 16 bytes, or Extended format as specified by the TimeFormat equipment constant value setting.	Where:  12-byte format YYMMDDhhmmss  16-byte format YYYYMMDDhhmmsscc  Extended (max 32 bytes) format YYYY-MM-DDThh:mm:ss.sTZD (see SEMI E148)  See TimeFormat equipment constant variable item in Table 4 for additional detail on the formats.	SV

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
ControlState	10, 51	This status variable contains the code which identifies the current control state of the equipment. When reported related to a control state transition, its value should represent the state current after the transition.	1 = OFF-LINE/EQUIPMENT OFF-LINE 2 = OFF-LINE/ATTEMPT ON-LINE 3 = OFF-LINE/HOST OFF-LINE 4 = ON-LINE/LOCAL 5 = ON-LINE/REMOTE 6-63 Reserved	SV
CycleCtr	5()	The number of machine cycles during the lifetime of the equipment. Non-resettable.		SV
DataSetName	20	The name of the Data Set that was uploaded through S7F37, F39, F41, F43; S15F49, F51 transactions that initiated Data Set transfer of large process programs and recipes.		DVVAL
DowntimeAlarm	3(), 5()	Identifier of the last alarm or exception triggering an equipment-initiated transition to UNSCHEDULED DOWNTIME from the PRODUCTIVE or STANDBY states.		SV
DowntimeAlarmText	20	Text associated with DowntimeAlarm. 0–80 characters.		SV
DowntimeData	20	Equipment defined data associated with transitions to, or within, the SCHEDULED or UNSCHEDULED DOWNTIME states. For example, this may be used to carry fault information, the component serial number of a repaired component, or comments entered at the equipment's control panel. 0–256 characters.		SV
EngTime	5()	Accumulation of time in ENGINEERING reported in minutes.		ECV
EqpModel	20	Text string describing the equipment model. 1–80 characters.		SV
EqpName	20	Text string containing a user-assigned name for equipment. 1–80 characters. Information in the data item EQNAME is a subset of EqpName.		ECV
EqpSerialNum	20	Text string describing the product serial number assigned by the manufacturer. 1–80 characters. Information in the data item MDLN is a subset of EqpSerialNum.		SV
EstablishCommunications Timeout	52	The length of time, in seconds, of the interval between attempts to send S1F13 when establishing communications.		ECV

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
EventsEnabled	0	Contains the list of events (CEIDs) enabled for reporting (via Stream 6).	Structure: L,n    n = # of events enabled 1.<CEID <sub>1</sub> > . . n.<CEID <sub>n</sub> >	SV
EventLimit	0, 10, 11, 20, 21, 3(), 4(), 5()	Used with the Limits Monitoring capability, it contains the LIMITID of the limit reached or crossed by LimitVariable. Since multiple zone transitions for a variable may occur simultaneously (e.g., due to identical limit definitions or a slow data sampling rate), EventLimit has been defined to allow for a list of LIMITIDs.		DVVAL
InterruptionCtr	5()	The number of transitions to UNSCHEDULED DOWNTIME from PRODUCTIVE.		ECV
LastPowerdown	20	Timestamp estimate of when the last powerdown or reset occurred. Uses format defined for CLOCK.		SV
LimitVariable		This variable contains the VID for the variable whose value changed monitoring zones.		DVVAL
MaxSpoolTransmit	54	The maximum number of messages which the equipment will transmit from the spool in response to an S6,F23 "Transmit Spooled Messages" request. If MaxSpoolTransmit is set to zero, no limit is placed on the messages sent from the spool. Multi-block inquire/grant messages are not counted in this total.		ECV
NSTime	5()	Accumulation of time in NON-SCHEDULED TIME, reported in minutes.		ECV
OperatorCommand	5()	This data variable is valid in the event the operator issues a command to the equipment. The codes for this variable are equipment dependent.		DVVAL
OverWriteSpool	11	This Equipment Constant is used to indicate to the equipment either to overwrite data in the spool area or to stop spooling whenever the spool area limits are exceeded.	= TRUE to overwrite spooled data = FALSE to stop spooling when limits exceeded	ECV



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
PowerdownTime	20	This timestamp is periodically updated based on an interval set by the user. It is used to determine the approximate time that the equipment went down in the event of a power loss. Uses format defined for CLOCK.		SV
PowerupState	20	Specifies the powerup ARAMS state when powerdown occurs during manufacturing time. Single text digit.	2 = STANDBY 5 = UNSCHEDULED DOWNTIME	SV
PPChangeName	10, 20	The PPID which was affected upon the event of the creation, editing, or deletion of a Process Program local to the equipment. If the PPID Data Item is also defined and implemented for the equipment, then the values for PPChangeName are subject to the same format restrictions defined for the PPID Data Item.		DVVAL
PPChangeStatus	51	The action taken on the Process Program named in PPChangeName. This variable is valid upon the event of the creation, editing, or deletion of a Process Program local to the equipment.	1 = Credited 2 = Edited 3 = Deleted 4-63 Reserved	DVVAL
PPError	20	Contains information about a failure to verify a text process program.		SV or DVVAL
PPExecName	0, 10, 20	The PPID(s) of the currently selected Process Program(s). The selection of a new Process Program updates this variable. If multiple Process Programs can be selected, then this variable is a list of PPIDs. If the PPID Data Item is also defined and implemented for the equipment, then the values for PPExecName are subject to the same format restrictions defined for the PPID Data Item.		SV

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
PPFormat	51	Indicates the type or types of process programs and recipes that are supported.	1 = Unformatted process programs 2 = Formatted process programs 3 = Both unformatted and formatted process programs 4 = Execution Recipes 5 = Large unformatted process programs 6 = Large formatted process programs 7 = Both large unformatted and large formatted process programs 8 = Large execution recipes 9 = Both execution recipes and large execution recipes 10 = Both unformatted and large unformatted process programs 11 = Both formatted and large formatted process programs 12 = Both unformatted and large formatted process programs 13 = Both formatted and large unformatted process programs 14 = Unformatted, large unformatted and large formatted process programs 15 = Formatted, large unformatted and large formatted process programs 16 = Formatted, unformatted, large unformatted, large formatted process programs >16 Reserved	SV
PrdRecovery	11	A boolean value that enables (TRUE) or disables (FALSE) the equipment-initiated return to PRODUCTIVE from UNSCHEDULED DOWNTIME.		ECV
PrdState	20	Default ARAMS Substate Code for automated transitions to PRODUCTIVE.		SV
PrdTime	5()	Accumulation of time in PRODUCTIVE, reported in minutes.		ECV
PrevARAMSState	20	The ARAMS code corresponding to the previous state/substate. Four characters.		SV
PreviousProcessState	51	The previous processing state of the equipment, before the most recent process state change.	0-63 Reserved	SV
ProcessState	51	The current processing state of the equipment.	0-63 Reserved	SV

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
RcpChangeName	20	The identifier of the recipe affected upon the event of the creation, editing, or deletion of a recipe.		DVVAL
RcpChangeStatus	51	The type of change that occurred for the recipe indicated in RcpChangeName.	0 = No change 1 = Created 2 = Updated (modified) 3 = Stored (new) 4 = Replaced 5 = Deleted 6 = Copied 7 = Renamed 8,9 Reserved >10 Reserved	DVVAL
RcpExecName	0, 20	The identifier, or a list of identifiers, of currently selected recipes. A zero-length item or list indicates no recipes are currently selected.		SV
ReticleIDVerification	11	This equipment constant is used to indicate whether Reticle ID verification is required.		ECV
ReticleParticleInspection	11	This equipment constant is used to indicate whether Reticle Particle Inspection is required.		ECV
SbyRecovery	11	A boolean value that enables (TRUE) or disables (FALSE) the equipment-initiated return to STANDBY from UNSCHEDULED DOWNTIME.		ECV
SbyTime	5()	Accumulation of time in STANDBY, reported in minutes.		ECV
SDTime	5()	Accumulation of time in SCHEDULED DOWNTIME, reported in minutes.		ECV
SpoolCountActual	5()	Used to keep a count of the messages actually contained in the equipment's spool area. Multi-block inquire/grant messages are not spooled and not included in this count.		SV
SpoolCountTotal	5()	Used to keep a count of the total number of primary messages directed to the spool, regardless of whether placed or retained in the spool. Multi-block inquire/grant messages are not spooled and not included in this count.		SV



<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
SpoolFullTime	20	Contains the timestamp from the time the spool last became full. If the spool was not filled during the last spooling period, this will contain a time value prior to the current SpoolStartTime. Uses the same format as the CLOCK variable data item.		SV
SpoolStartTime	20	Contains the timestamp from the time spooling last became active. Uses the same format as the CLOCK variable data item.		SV
SymptomID	5()	A numeric code representing the symptom that initiated the user-initiated state change. A value of zero indicates "no symptom."		SV
SymptomText	20	Text describing the SymptomID. 0–80 characters.		SV

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
TimeFormat	5()	The setting of this ECV controls whether the equipment shall use the variable item CLOCK and the data items STIME, TIMESTAMP, and TIME in 12-byte, 16-byte, or Extended format.	<p>Where:</p> <p>0 = 12-byte format  1 = 16-byte format  2 = Extended (max 32 byte) format  &gt;2 = Reserved</p> <p>12 byte format  YYMMDDhhmmss  YY = Year                      00 to 99  MM = Month                    01 to 12  DD = Day                      01 to 31  hh = Hour                      00 to 23  mm = Minute                   00 to 59  ss = Second                    00 to 59</p> <p>16 byte format  YYYYMMDDhhsscc  YYYY = Year                   0000 to 9999  MM = Month                   01 to 12  DD = Day                      01 to 31  hh = Hour                      00 to 23  mm = Minute                   00 to 59  ss = Second                   00 to 59  cc = Centisecond              00 to 99</p> <p>Extended (max 32 byte) format  YYYY-MM-DDThh:mm:ss.sTZD  (see SEMI E148)  YYYY = Year                   0000 to 9999  MM = Month                   01 to 12  DD = Day                      01 to 31  T = special separator        T  hh = Hour                      00 to 23  mm = Minute                   00 to 59  ss = Second                   00 to 59  s = one to six digits  representing a fraction of a second  TZD = time zone designator  ('+hh:mm' or '-hh:mm' for local time and 'Z' for UTC time)  Example:  1985-04-12T14:15:30.450Z  (UTC)  1985-04-12T10:15:30.450-04:00 (local)</p>	ECV
TransitionType	10	Used with the Limits Monitoring capability, it defines the direction of the zone transition which has occurred.	<p>0 = Transition from lower to upper zone.  1 = Transition from upper to lower zone.</p>	DVVAL
TRATID	5()	Contains the TRATOMICID of the atomic transfer referenced by the event.		DVVAL

<i>Name</i>	<i>Format</i>	<i>Description</i>	<i>Values</i>	<i>Class</i>
TRJOBIDENT	5()	Contains the TRJOBID for the transfer job referenced by the event.		DVVAL
TRJOBNM	20	Contains the TRJOBNAME for the transfer job referenced by the event.		DVVAL
TRLNK	5()	Contains the TRLNK value for the atomic transfer referenced by the event.		DVVAL
UDTime	5()	Accumulation of time in UNSCHEDULED DOWNTIME, reported in minutes.		ECV

9.8 *Object Dictionary* — This section defines the public attributes of objects which are available through SECS-II messages.

9.8.1 The attributes of an object are defined in a table for each object in the following form:

**Table 5 Agent Attribute Definition**

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	Agent object type.	RO	20	“Agent”
“ObjID”	The agent’s name, assigned by an <i>authorized user</i> .	RO	20	

**Table 6 Agent-Specific Dataset Object Attribute Definition**

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“MRcpASDS”
“ObjID”	The object’s identifier. Contains the value in <u>AgentSpec Agent</u> .	RO	20	
“AgentSpec_Agent”	The name of the <i>executing agent</i> to which the other attributes in the <i>dataset</i> apply. <i>Mandatory</i> .	RO	20	
“AgentSpec_AttrLength”	The length of the <i>agent-specific</i> attributes, in bytes. <i>Mandatory</i> .	RO	54	
“AgentSpec_ChgTime”	Timestamp of when an <i>agent-specific</i> attribute was last changed. <i>Mandatory</i> .	RO	20	
“AgentSpec_Comments”	Comments specific to the <i>agent</i> entered by the author.	RW	20	Maximum length is 80 characters.
“AgentSpec_LinkParam”	A list of <i>variable parameter definitions</i> modified from the list in <u>LinkParam</u> . Valid only for a <i>linked main</i> recipe. <i>Parameter name</i> and form may not be changed.	RO	00	List of structures composed of parameter name, value, and restrictions.
“Certified”	The certification level for the specific <i>agent</i> , assigned by an <i>authorized user</i> . Reset when <u>AgentSpec LinkParam</u> is modified. Required for <i>certification</i> support.	RW	52	
“AgentSpec_UD_”	Non-standard attribute defined by the supplier or user. Asterisk indicates the part of the attribute name that is provided in this definition. Must be preserved exactly except by the defining entity.	RO	10, 11 20, 3() 4(), 5()	Text form is limited to 80 characters.

**Table 7 Collection Event Object**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Items</i>	<i>Value</i>
“ObjType”	Collection Event Object type.	RO	20	-	“COLLEVENT”
“ObjID”	Collection Event Identifiers.	RO	20	CEID	-
“Enabled”	Boolean true means reporting is enabled for a specific CEID.	RW	11	CEED	-
“EventSource”	Object specifier for object which generates the event for a specific CEID.	RO	20	EVNTSRC	-
“DataReportList”	List of Report Identifiers linked to a specific CEID.	RO	20	-	(list of) RPTID

**Table 8 Data Report Object**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Items</i>	<i>Value</i>
“ObjType”	Data Report Object Type.	RO	20	-	“DATARPT”
“ObjID”	Object identifier for a data report.	RO	20	RPTID	-
“DataSource”	Source for the variable data, not writable for predefined provider reports.	RO	20	DATASRC	-
“AttrList”	Returns the attribute (or variable) names that this report is requesting from the Data source.	RW	*	-	(list of) VID

**Table 9 Data Source Object**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Items</i>	<i>Value</i>
“ObjType”	Data Source Object Type.	RO	20	-	“DataSource”
“ObjID”	Identifier of a specific Data Source Object.	RO	20	DATASRC	-
“AttrList”	Name of attributes for a specific Data Source Object.	RO	*	-	(list of) VID

**Table 10 Distributed Recipe Namespace Attribute Definition**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“RNSD”
“ObjID”	Text.	RO	20	
“LockedRecipes”	A list of <i>identifiers</i> of all recipes with existing <i>change request records</i> .	RO	20	
“Recorder”	The <i>recorder specifier</i> of the attached <i>distributed recipe namespace recorder</i> .	RO	20	
“Segments”	A list of <i>specifiers</i> of the <i>distributed recipe namespace segments</i> attached to the <i>namespace</i> .	RO	20	

**Table 11 Distributed Recipe Namespace Manager Attribute Definition**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“RNS_MgrD”
“ObjID”	The <i>manager’s</i> name.	RO	20	

**Table 12 Distributed Recipe Namespace Recorder Attribute Definition**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“RNSDRecorder”
“ObjID”	Text.	RO	20	
“LockedRecipes”	List of <i>identifiers</i> of recipes with existing <i>change request records</i> .	RO	00	
“Namespace”	Identifies the <i>namespace</i> to which the recorder is attached. May be set by the manager.	RO	20	
“NamespaceManager”	Identifies the <i>distributed recipe namespace manager</i> . May be set by the manager.	RO	20	
“Segments”	List of <i>specifiers</i> of currently attached segments.	RO	00	

**Table 13 Distributed Recipe Namespace Segment Attribute Definition**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“RNSDSegment”
“ObjID”	The object name (identifier).	RO	20	
“Namespace”	The name (ObjID) of the <i>namespace</i> to which the <i>segment</i> belongs. May be set by the manager.	RO	20	
“NamespaceManager”	Identifies the <i>distributed recipe namespace manager</i> . May be set by the manager.	RO	20	
“RecipeReadOnlyLevel”	Used to track the corresponding attribute of the <i>namespace</i> to which the <i>segment</i> belongs. May be set by the manager.	RO	52	

**Table 14 Exception Attributes**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Items</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	-	“EXCEPTION”
“ObjID”	The identifier of a specific Exception.	RO	20	EXID	-
“EXType”	Identifies the type of exception.	RO	20	-	Select from set: “ALARM” “ERROR”
“EXMessage”	Text message describing the abnormal situation monitored.	RO	20	-	Max. length of 80 characters
“EXEnabled”	Indicates that reporting to the decision authority on the exception condition is enabled.	RW	11	-	Boolean; TRUE is enabled.
“EXRecoveryAction”	List of possible recovery actions (EXRecovery).	RO	20	-	List of text
“EXState”	Current state of an Exception Object. The Exception Object state is usually defined as a combination of substates and concurrent states.	RO	20	-	Composed from the set: “CLEARED” “SET” “NOTPOSTED” “POSTED” “NOTRECOVERING” “RECOVERING” “ABORTINGRECOVERY”



**Table 15 Execution Recipe Attribute Definition**

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Rqmt</i>	<i>Form</i>	<i>Default Value</i>
<i>Identification Attributes</i>					
“ObjType”	The object type.	RO	Y	Text: “ERcp”	“ERcp”
“ObjID”	An identifier derived from Namespace, Class, Name, and Version.	RO	Y	Formatted text.	-
“Namespace”	The name of the <i>originating namespace</i> .	RO	Y	Text.	-
“Name”	A logical name assigned by the user when the recipe is <i>created</i> .	RO	Y	Text.	-
“Class”	The recipe’s class (e.g., “/PROCESS/” OR “/PROCESS/LOADER/”).	RO	Y	Formatted text: “CLASS/CLASS/ /CLASS/”	-
“Version”	The version of the recipe.	RO	Y	Text.	-
<i>Mandatory Attributes</i>					
“ExecAttrLength”	The <i>length attribute</i> for the attributes of the <i>execution recipe</i> . Calculated when the recipe is <i>downloaded</i> and whenever an attribute changes. <i>Mandatory</i> .	RO	Y	Unsigned integer.	-
“ExecChgTime”	The <i>timestamp</i> of a change to the attributes of the <i>execution recipe</i> . <i>Mandatory</i> .	RO	Y	Formatted text, <i>timestamp</i> format	-
“AttrLength”	Preserved. <i>Mandatory</i> .	RO	Y	Unsigned integer.	-
“AttrChgTime”	Preserved. <i>Mandatory</i> .	RO	Y	Timestamp format.	-
“EditTime”	Preserved unless recipe is modified. <i>Timestamp</i> of when the <i>body</i> was <i>created</i> or modified. <i>Mandatory</i> .	RO	Y	Formatted text <i>Timestamp</i> format	-
“BodyLength”	Preserved unless recipe is modified. Length of the recipe’s body, in bytes. <i>Mandatory</i> .	RO	Y	Unsigned integer.	-
“BodyFormat”	Indicates the form and format of the recipe’s <i>body</i> .	RO	Y	Enumerated unsigned integer: 0 = <i>source</i> , 1 = <i>object</i> , > 1 reserved.	0
“Verified”	Indicates whether the recipe’s body is syntactically correct.	RO	Y	Boolean.	FALSE
“Linked”	Indicates whether the recipe is <i>linked</i> .	RO	Y	Boolean.	FALSE
“ChangedBody”	Set to TRUE if the recipe body has changed without a subsequent upload to the originating namespace. Note: this attribute is never updated to a namespace. Required only if recipe can be changed or created.	RO	Y	Boolean.	FALSE

Attribute Name	Definition	Access	Rqmt	Form	Default Value
“ExecChgCtl”	Preserved. Specifies change control requirements for recipe.	RO	Y	Binary. Bitwise: 1 – may change 2 – change notification required 4 – recipe may be selected after change, 8 – most recent parameter settings shall be saved. Any combination of these four bits is allowed.	0
<i>Optional Attributes</i>					
“AgentSpec_Comments”	Copied from the original <i>agent-specific</i> attribute when downloaded. Set by the user.	RO	N	Text. Maximum length is 80 characters.	-
“ApprovalLevel”	Indicates the level of approval assigned by an <i>authorized</i> user.	RO	N	Unsigned integer.	0
“Certified”	Preserved from the <i>agent-specific</i> attribute as downloaded. May be used as control for production-worthy recipes.	RO	N	Unsigned integer.	0
“Comments”	User comments. Preserved from the <i>generic</i> attribute as downloaded.	RO	N	Text. Maximum length is 80 characters.	-
“EditedBy”	Preserved unless recipe is modified. The name of the person or <i>executing agent</i> who last modified the recipe.	RO	N	Text. Maximum length is 40 characters.	-
“EstRunTime”	The nominal or estimated execution (run) time of the recipe, in seconds. Used for scheduling purposes. Preserved from the <i>generic</i> attribute as downloaded.	RO	N	Unsigned integer.	0
“ExecLinkParam”	Preserved unless last value is changed (§ 9.7.4). Contains the list of <i>parameter definitions</i> including any <i>agent-specific</i> modifications. Required for <i>variable parameter</i> support.	RO	N	Structure composed of parameter name, initial value, and restrictions.	NULL
“LinkList”	Preserved. A complete list of recipe <i>specifiers</i> for a <i>linked recipe set</i> . Required for multi-part recipe support.	RO	N	List of formatted text.	NULL
“SrcRcpID”	For a derived <i>object form</i> recipe, contains the recipe <i>identifier</i> of the original <i>source form</i> recipe. Required only for <i>derived object form</i> recipes.	RO	N	Formatted text.	NULL
“VerificationID”	Identification code used by the <i>verifier</i> of the recipe. May be used to determine out-of-date formats that need to be <i>re-verified</i> .	RO	N	Text. Maximum length is 40 characters.	NULL
<i>Non-Standard Attributes</i>					
AgentSpec_UD_*	Preserved from the original <i>agent-specific</i> attributed as downloaded.	RO	N	Defined by supplier or user. Text limited to 80 characters.	-

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Rqmt</i>	<i>Form</i>	<i>Default Value</i>
UD_*	Non-standard attribute defined by supplier or <i>user</i> . Asterisk indicates the part of the attribute name that is provided in this definition. Shall be preserved exactly except by the entity that defined it.	RO	N	Varies with definition. Text form is limited to 80 characters.	-

**Table 16 Managed Recipe Attribute Definition**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“MRcp”
“ObjID”	An identifier derived from <u>Class</u> , <u>Name</u> , and <u>Version</u> . No part of a recipe’s identifier shall be changed except through <i>renaming</i> .	RO	20	
(other)	Description of the information contained.	RO or RW	Varies with definition.	Varies with definition.
“Name”	A logical name assigned by the user when the recipe is <i>created</i> or <i>renamed</i> .	RO	20	
“Class”	The recipe’s class (e.g., “/PROCESS/” or “/PROCESS/LOADER/”).	RO	20	Formatted text: “CLASS/CLASS/./CLASS/”
“Version”	The version of the recipe.	RO	20	
“AttrLength”	The total length of the <i>generic</i> attributes, in bytes. <i>Mandatory</i> .	RO	5()	
“AttrChgTime”	Timestamp of the last change to a <i>generic</i> attribute. <i>Mandatory</i> .	RO	20	
“BodyLength”	Length of the recipe’s body, in bytes. <i>Mandatory</i> .	RO	5()	
“EditTime”	Timestamp of when the <i>body</i> was <i>created</i> or last <i>updated</i> . <i>Mandatory</i> .	RO	20	<i>Timestamp</i> format: “YYYYMMDDhhmmsscc”
“BodyFormat”	Indicates the form and format of the recipe’s <i>body</i> . <i>Default</i> is zero.	RO	52	0 = source, 1 = object, >1 reserved.
“Verified”	Indicates whether the recipe’s body is syntactically correct. Reset when the recipe is <i>created</i> or <i>updated</i> . <i>Default</i> is FALSE.	RO	11	
“Linked”	Indicates whether the recipe is <u>linked</u> . Reset when the recipe is <i>originated</i> , <i>verified</i> , or <i>unlinked</i> . <i>Default</i> is FALSE.	RO	11	
“ApprovalLevel”	Indicates the level of approval assigned by an <i>authorized user</i> . <i>Default</i> is zero. Reset when the recipe is <i>originated</i> or <i>linked</i> . For a <i>linked</i> recipe, may not be higher than any of its <i>subrecipes</i> .	RW	52	
“Comments”	User comments.	RW	20	Maximum length is 80 characters.
“EditedBy”	The name of the person who last edited the recipe.	RO	20	Maximum length is 40 characters.
“EstRunTime”	The nominal or estimated execution (run) time of the recipe, in seconds. Reset when the recipe, is <i>created</i> or <i>updated</i> . Set when the recipe is <i>verified</i> . May be recalculated to total time for a <i>main</i> recipe when <i>linked</i> . Used for scheduling purposes. Algorithm for calculation shall be documented. <i>Default</i> is 0.	RW	54	

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
"ExecChgCtrl"	Specifies change control requirements for recipe. Default is 0. Combinations of bits are used to indicate multiple permissions.	RW	10	Binary. Bitwise (MSB = 8): 1 – the recipe body may be changed 2 – change notification required 4 – recipe may be selected after change, 8 – most recent parameter settings shall be saved. Any combination of these four bits is allowed
"ExtRef"	A list of all recipe <i>specifiers</i> as referenced within the recipe. Explicit <i>versions</i> not required. Reset when the recipe is <i>created</i> , <i>updated</i> , and <i>verified</i> .	RO	00	List of items of format 20.
"LinkList"	A complete list of recipe <i>specifiers</i> found in the <u>ExtRef</u> attribute of a <i>main</i> recipe and all of its <i>subrecipes</i> , with duplicates removed and all <i>versions</i> explicitly determined. Set for the <i>main</i> recipe when <u>linked</u> . Reset when the recipe is <i>originated</i> or <i>verified</i> . Required for multi-part recipe support.	RO	00	List of items of format 20.
"LinkParam"	A list of all variable parameter definitions contained in the <u>Parameters</u> attribute of a <i>main</i> recipe and all of its <i>subrecipes</i> , with duplicates removed. Reset when the recipe is <i>created</i> , <i>updated</i> , or <i>verified</i> . Set when the recipe is <i>linked</i> . Required for <i>variable parameter</i> support.	RO	00	List of parameter definition structures composed of parameter name, initial value, and restrictions.
"Parameters"	A list of variable parameter definitions contained in the recipe. Reset when the recipe is <i>created</i> , <i>updated</i> , and <i>verified</i> . Set when the recipe is <i>verified</i> . Required only for <i>variable parameter</i> support.	RO	00	List of parameter definition structures composed of parameter name, initial value, and restrictions.
"SrcRecID"	<i>Identifier</i> of the <i>source form</i> recipe from which a <i>derived object form</i> recipe is derived. Value determined by the <i>verifier</i> of the recipe. Required only for support of <i>derived object form</i> recipes.	RO	20	
"VerificationID"	Identification code set by the <i>verifier</i> of the recipe. May be used to determine out-of-date formats that need to be <i>re-verified</i> .	RO	20	Maximum length is 40 characters.
"UD_"	Non-standard attribute defined by supplier or <i>user</i> . Asterisk indicates the part of the attribute name that is provided in this definition. Shall be preserved exactly except by the entity that defined it.	RO	10, 11, 20, 3(), 4(), 5()	Text form is limited to a maximum of 80 characters.

**Table 17 Process Job Attributes**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Item</i>	<i>Value</i>
"ObjType"	Name of the Object Type.	RO	20	-	"PROCESSJOB"
"ObjID"	Identifier of a Process Job.	RO	20	PRJOBID	-
"PRMtlType"	Type of material being processed.	RO	20	-	allowed values: "css" "wfr"

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Item</i>	<i>Value</i>
"PRMt1NameList"	Process Material Name, identifies material being processed by a job, which could be more than one item.	RO	20	-	(list of) Text
"RecID"	Object Specifier of Recipe used by a Process Job, see SEMI E39 and SEMI E42.	RO	20	-	-
"PRRecipeMethod"	Indicates any special handling for a Process Job's Recipe.	RO	20	-	allowed values: "STANDARD" "USETUNING"
"PRJobState"	Indicates the current state of a Process Job. The state of a job may be a combination of sub-states and concurrent states.	RO	20	-	Composed from the set: "WAITINGFOR JOB" "JOBQUEUED" "JOB CANCELLED" "JOBACTIVE" "SETUP" "WAITINGFORSTART" "PROCESSING" "NOTPAUSED" "PAUSING" "PAUSED" "NOTSTOPPING" "STOPPING" "NOTABORTING" "ABORTING" "PROCESSCOMPLETE" "JOBCOMPLETE"
"PRProcessStart"	Processing should start automatically after Job is defined when this Boolean is set TRUE.	RO	11	-	Boolean

**Table 18 Recipe Executor Attribute Definition**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Rqmt</i>	<i>Form</i>
"ObjType"	The object type.	RO	Y	Text = "RcpExec"
"ObjID"	Text.	RO	Y	Text
"DefaultNamespace"	The name of an <i>executing agent's name-space</i> used for all hardware-dependent and other <i>agent-specific</i> recipes.	RW	Y	Text
"ProdApprove"	The minimum value of a recipe's <i>approval level</i> accepted during productive and standby states. Required for SEMI E10 support only.	RW	N	Unsigned integer
"ProdCertify"	The minimum value of a recipe's <i>certification level</i> accepted during productive and standby states. Required for SEMI E10 support only.	RW	N	Unsigned integer
"RunCycleUnit"	The process unit on which the calculation of the estimated value of the recipe <i>generic attribute</i> EstRunTime is based.	RO	N	Case-sensitive formatted text composed of a unit of measure and an optional numeric suffix. Compliant with SEMI E5, § 12.
"RecipeSelectID"	A list of recipe <i>identifiers</i> for the currently selected recipes.	RO	Y	List of formatted text.

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Rqmt</i>	<i>Form</i>
“RecipeSelect-Parameters”	A list of all <i>parameter definitions</i> in effect for the <i>i</i> th recipe <i>identifier</i> in RecipeSelectID. The maximum value for <i>i</i> is determined by the equipment supplier as the maximum number of recipes which may be <i>selected</i> at the same time. Required if variable parameters are supported.	RO	N	List of structures composed of parameter name, parameter value, parameter restriction.

**Table 19 Recipe Namespace Attribute Definition**

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“RNS”
“ObjID”	The <i>name</i> of the <i>namespace</i> .	RO	20	A name of “Default” is prohibited.
“RecipeReadOnlyLevel”	The level of <i>approval</i> at which recipes are <i>read-only</i> .	RW	52	
“Members”	The <i>names</i> of <i>agents</i> capable of <i>verifying</i> and <i>executing</i> the recipes in the <i>namespace</i> .	RW	00	List of items of format 20.

**Table 20 Recipe Namespace Manager Attribute Definition**

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“RNS_Mgr”
“ObjID”	The <i>manager’s</i> name.	RO	20	
“NamespaceName”	The <i>name</i> of the <i>namespace</i> managed.	RO	20	

**Table 21 Table Attribute Definition**

<i>Attribute Name</i>	<i>Definition</i>	<i>Access</i>	<i>Format</i>	<i>Value</i>
“ObjType”	The object type.	RO	20	“Table”
“ObjID”	The object’s identifier.	RO	20	1–80 characters.
“NumCols”	Number of columns.	RO	5()	Non-zero.
“NumRows”	Number of rows.	RO	5()	Non-zero.
“TableLength”	Total number of bytes required to store the table elements, exclusive of any formatting required for storage.	RO	5()	Non-zero.

**Table 22 Trace Object**

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Items</i>	<i>Value</i>
“ObjType”	Trace Report Object type.	RO	20	-	“TRACE”
“ObjID”	Identifier of a specific Trace Report.	RO	20	TRID	-
“Enabled”	Boolean true means the specific Trace Report is enabled.	RW	*	CEED	-
“ReportID”	List or report linked to this Trace Report.	RO	20	-	(list of) RPTID
“SamplePeriod”	Time between report samples given in floating point seconds.	RW	4 ()	TRSPER	-
“TotalSamples”	The maximum number of samples that this Trace Report will perform.	RW	*	TOTSMP	-
“GroupSize”	Number of trace reports to be grouped before a report is sent.	RW	*	REPGSZ	-

<i>Attribute Name</i>	<i>Description</i>	<i>Access</i>	<i>Format</i>	<i>Related Data Items</i>	<i>Value</i>
“StartEventID”	Identifier of the event which starts trace reporting.	RW	20	CEID	-
“StartEvtSrcSpec”	Source for the start event.	RW	20	EVNTSRC	-
“StopEventID”	Identifier of the event which stops trace reporting.	RW	20	CEID	-
“StopEvtSrcSpec”	Source for the stop event.	RW	20	EVNTSRC	-
“AutoDelete”	Boolean true means this report is deleted when reporting is complete.	RW	11	TRAUTOD	-
“ReportChangeOnly”	Boolean, if true, then trace reports are sent only if at least one of the reported variables changes.	RW	11	RPTOC	-

9.8.2 *Attribute Name* — A reserved text string, of at most 40 characters, that is unique for that object.

9.8.3 *Description* — A description of the attribute.

9.8.4 *Access* — Indicates whether the attribute may be set through messages. Access is either read-only (RO) or read-write (RW).

9.8.5 *Format* — Indicates the type of data (format code).

9.8.6 *Timestamp Format* — Text form indicating date and time in the format as specified by the TimeFormat equipment constant value setting.

9.8.7 *Related Data Items* — Indicates an explicit relationship with a corresponding data item.

9.8.8 *Value* — Specifies any restrictions on the possible values. Examples of restrictions include exclusion of zero for format 5(), a maximum length for text, a format imposed on text, an order imposed on a list, or an enumerated set of valid values.

9.8.9 *Requirements*

- The attributes “ObjType” and “ObjID” are required for all object definitions and shall use format 20.
- The attribute “ObjType” shall be assigned a fixed value for each object.
- The value of “ObjID” may not be changed by using SetAttr (S14F3).

9.8.10 The value of “ObjType” may be used for messages using the data item OBJTYPE. The value of the attribute “ObjID” may be used for messages using the data item OBJID.

9.8.11 The name of a public attribute may be used for messages using the data item ATTRID. The value of a public attribute may be used for messages using the data item ATTRDATA.

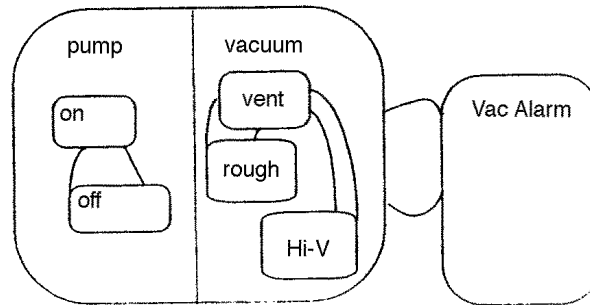
9.8.12 Variable data items defined in § 9.7 may be regarded as attributes of the object type “Equipment”, where SVs and DVVALs are RO and ECVs are RW.

9.9 With the use of Harel<sup>4</sup> state diagrams to describe the behavior of objects, an object’s state must be describable as a combination of a set of sub-states and concurrent states. The rules for describing the state of an object are: (1) use the comma (‘,’) to delimitate concurrent states, (2) use the foreslash (‘/’) to delimit a super-state and sub-state, (3) to deliver the set of lowest level concurrent states, and (4) optionally omit super-state names when there are no ambiguities in the names of the lowest level states.

9.9.1 Please refer to Figure 3 in order to follow the discussions for the notations. In Harel notation, ‘pump’ and ‘vacuum’ are concurrent states. The text to specify this relation in a response to a request for state is ‘pump, vacuum’. The comma can be read as meaning ‘and’. ‘on’ and ‘off’ are sub-states of ‘pump’. ‘vent’, ‘rough’, and ‘Hi-V’ are substates of ‘vacuum’. The substate syntax is ‘state/sub-state’ where the ‘/’ can be read as ‘is in sub-state’.

4 Harel D. “Statecharts: A Visual Formalism for Complex Systems,” Science of Computer Programming, 8, 1987, pp. 231-274. Elsevier Science, P.O. Box 945, New York, NY 10159-0945, USA; <http://www.elsevier.nl/homepage/browse.htm>

So using the example in Figure 3, if the pump is off and the vacuum is vented, then the text message which conveys this is 'pump/off, vacuum/vent'. This message can be shortened to 'off,vent' because there is no ambiguity in doing so.



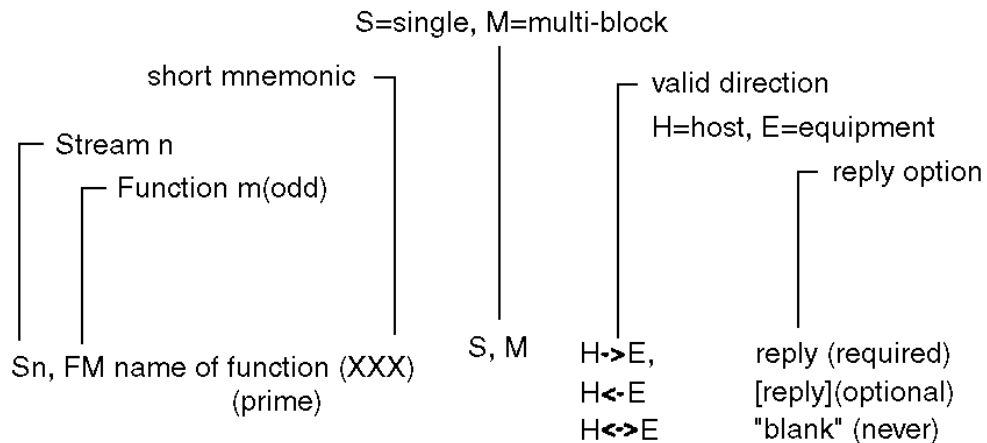
**Figure 3**  
**State Chart Example**

## 10 Message Detail

10.1 *Intent* — This section defines a number of specific functions in different streams which can be used as a basis for communication between host and equipment. The functions are defined in the form of transaction message pairs according to the transaction level requirements specified in § 8.

10.2 The functions are described in a standard form which involves specification of the number, name, single or multiple block, direction of communication, nature of reply required, description, variable definition, and the detailed structure of the message in terms of lists and items. Double lines separate streams, and single lines separate transactions to aid readability.

10.2.1 The abbreviations used in each transaction are as follows:





<i>Description</i>
A description of the action generated by the function.
<i>Structure</i>
Detailed structure showing lists and defined items. Lists are denoted by a capital L followed by the length separated by a comma. The individual elements in the list are numbered on separate lines. Nested lists are indented to emphasize the structure. The detailed form of the items is given in the define section at the beginning of the transaction. The symbols "<" and ">" are used to enclose each item in the structure data and imply that there is an item header. A detailed description of each data item as well as a list of the allowable data formats can be found in the Data Item Dictionary.
<i>Exception</i>
Special cases in the structure that have a different meaning.

Sn,Fm+1 Name of function (secondary) (same structure as above except never with reply)

10.3 *Message Usage* — This section discusses message features and where they may be used.

10.3.1 *Zero Length Items and Lists* — Certain message definitions may use zero length data items and zero length lists as a technique to convey specific information to the receiver of the message. For commands (i.e., “Do Something”) and requests (i.e., “Return Some Data”), it may be used to mean “Use default values for the data item(s) which were not included.” The default may be a specific value or a value chosen by the equipment.

10.3.1.1 For messages reporting data (either responses to requests or asynchronous reports), the technique may be used to indicate that the desired information is not available or not applicable. In some cases, the fact that data is unavailable may indicate success or failure of a command.

10.3.1.2 Certain message definitions may define a zero length data item or a zero length list to mean “the information is not supplied.” The receiving party should react to this lack of information as it deems appropriate.

10.3.1.3 *Compliance to Message Definitions* — Any given standard SECS-II message shall comply to the format shown in the Message Definition for that Stream and Function. Specifically:

1. The message shall contain all Lists and Data Items shown as required in the Message Definition.
2. The message shall not contain any Lists or Data Items not shown in the Message Definition, unless the Message Definition specifically allows this.
3. The message shall not contain any List Item or Data Item with zero length unless the Message Definition specifically defines a meaning for such a zero length item.

10.4 *Stream 0 and Function 0* — Stream 0 is always defined as not used since a 0 is the most likely error. No functions are defined in stream 0.

10.4.1 Function 0 exists in all streams and has the same special meaning in each stream. A function 0 message closes a transaction, so that the originator will not have to wait for a transaction timeout to proceed. Function 0 is sent in lieu of the expected secondary message when the interpreter cannot, because of a transmission error or some other reason, respond with the expected reply. It is not a requirement that the interpreter send function 0 to close a transaction.

10.5 *Stream 1 Equipment Status* — This stream provides a means for exchanging information about the status of the equipment, including its current mode, depletion of various consumable items, and the status of transfer operations.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F0 Abort Transaction (S1F0)	S,H<->E
<i>Description</i>	
Used in lieu of an expected reply to abort a transaction. Function 0 is defined in every stream and has the same meaning in every stream.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F1 Are You There Request (R)	S,H<->E,reply
<i>Description</i>	
Establishes if the equipment is on-line. A function 0 response to this message means the communication is inoperative. In the equipment, a function 0 is equivalent to a timeout on the receive timer after issuing S1,F1 to the host.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F2 On Line Data (D)	S,H<->E
<i>Description</i>	
Data signifying that the equipment is alive.	
<i>Structure</i>	
L, 2 1. <MDLN> 2. <SOFTREV>	
<i>Exception</i>	
The host sends a zero-length list to the equipment.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F3 Selected Equipment Status Request (SSR)	S,H->E,reply
<i>Description</i>	
A request to the equipment to report selected values of its status.	
<i>Structure</i>	
The following structure is approved for all item formats and should be used by all new implementations:  L, n 1. <SVID <sub>1</sub> > . . n. <SVID <sub>n</sub> >  The following structure is included for compatibility with previous implementations and may only be used for items of format 3() and 5():  <SVID <sub>1</sub> , ..., SVID <sub>n</sub> >	

<i>Exception</i>	
A zero-length list (structure 1) or item (structure 2) means report all SVIDs.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F4 Selected Equipment Status Data (SSD)	M,H<-E
<i>Description</i>	
The equipment reports the value of each SVID requested in the order requested. The host remembers the names of values requested.	
<i>Structure</i>	
L, n 1. <SV <sub>1</sub> > . . n. <SV <sub>n</sub> >	
<i>Exception</i>	
A zero-length list item for SV <sub>i</sub> means that SVID <sub>i</sub> does not exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F5 Formatted Status Request (FSR)	S,H->E,reply
<i>Description</i>	
A request for the equipment to report the status according to a pre-defined fixed format.	
<i>Structure</i>	
<SFCD>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F6 Formatted Status Data (FSD)	M,H<-E
<i>Description</i>	
The equipment reports the value of status variables according to the SFCD.	
<i>Structure</i>	
Depends upon the structure specified by the status form.	
<i>Exception</i>	
A zero-length item means that no report can be made.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F7 Fixed Form Request (FFR)	S,H->E,reply
<i>Description</i>	
A request for the form used in S1,F6.	
<i>Structure</i>	
<SFCD>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F8 Fixed Form Data (FFD)	M,H<-E
<i>Description</i>	
The form is returned with the name of each value and the data format item having a zero length as a two-element list in the place of each single item to be returned in S1,F6.	
<i>Structure</i>	
Depends upon the form being specified.	
<i>Exception</i>	
A zero-length item means the form is unavailable.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F9 Material Transfer Status Request (TSR)	S,H->E,reply
<i>Description</i>	
A request to report the status of all material ports to the host.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F10 Material Transfer Status Data (TSD)	M,H<-E
<i>Description</i>	
The equipment reports to the host the transfer status of all material ports.	
<i>Structure</i>	
L, 2 1. <TSIP <sub>1</sub> , . . . , TSIP <sub>n</sub> > 2. <TSOP <sub>1</sub> , . . . , TSOP <sub>n</sub> >	
<i>Exception</i>	
A zero-length item means there are no such ports. A zero-length list means there are no ports.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F11 Status Variable Namelist Request (SVNR)	S,H->E,reply
<i>Description</i>	
A request to the equipment to identify certain status variables.	
<i>Structure</i>	
L, n 1. <SVID <sub>1</sub> > . . n. <SVID <sub>n</sub> >	
<i>Exception</i>	
A zero length means report all SVIDs.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F12 Status Variable Namelist Reply (SVNRR)	M,H<-E
<i>Description</i>	
The equipment reports to the host the name and units of the requested SVs.	
<i>Structure</i>	
L, n 1. L, 3 1. <SVID <sub>1</sub> > 2. <SVNAME <sub>1</sub> > 3. <UNITS <sub>1</sub> > 2. L, 3 . . n. L, 3 1. <SVID <sub>n</sub> > 2. <SVNAME <sub>n</sub> > 3. <UNITS <sub>n</sub> >	
<i>Exception</i>	
Zero-length ASCII items for both SVNAME <sub>i</sub> and UNITS <sub>i</sub> indicates that the SVID does not exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F13 Establish Communications Request (CR)	S,H<->E,reply
<i>Description</i>	
The purpose of this message is to provide a formal means of initializing communications at a logical application level both on power-up and following a break in communications. It should be the following any period where host and Equipment SECS applications are unable to communicate. An attempt to send an Establish Communications Request (S1,F13) should be repeated at programmable intervals until an Establish Communications Acknowledge (S1,F14) is received within the transaction timeout period with an acknowledgement code accepting the establishment.	
<i>Structure</i>	
L, 2 1. <MDLN> 2. <SOFTREV>	
<i>Exception</i>	
The host sends a zero-length list to the equipment.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F14 Establish Communications Request Acknowledge (CRA)	S,H<->E
<i>Description</i>	
Accept or deny Establish Communications Request (S1,F13). MDLN and SOFTREV are on-line data and are valid only if COMMACK = 0.	
<i>Structure</i>	
L, 2 1. <COMMACK> 2. L, 2 1. <MDLN> 2. <SOFTREV>	
<i>Exception</i>	
The host sends a zero-length list for item 2 to the equipment.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F15 Request OFF-LINE (ROFL)	S,H->E,reply
<i>Description</i>	
The host requests that the equipment transition to the OFF-LINE state.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F16 OFF-LINE Acknowledge (OFLA)	S,H<-E
<i>Description</i>	
Acknowledge or error	
<i>Structure</i>	
<OFLACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F17 Request ON-LINE (RONL)	S,H->E,reply
<i>Description</i>	
The host requests that the equipment transition to the ON-LINE state	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F18 ON-LINE Acknowledge (ONLA)	S,H<-E
<i>Description</i>	
Acknowledge or error	
<i>Structure</i>	
<ONLACK>	
<i>Exception</i>	
None	

### 10.5.1 Macro Level Messages

<i>Stream,Function Name (Mnemonic)</i>		<i>Direction</i>
S1,F19 Get Attribute (GA)		S,H<—>E,reply <sup>5</sup>
<i>Description</i>		
Request for attribute data relating to the specified object or entity within the equipment.		
<i>Structure</i>		
L, 3 1. <OBJTYPE> 2. L,m [m = number of objects for which attributes requested] 1. <OBJID <sub>1</sub> > . . m. <OBJID <sub>m</sub> > 3. L,n [n = number of attributes requested for each object] 1. <ATTRID <sub>1</sub> > . . n. <ATTRID <sub>n</sub> >		
<i>Exception</i>		
A zero-length list (m = 0) is a request for attributes of all objects of the specified type. A zero-length list (n = 0) is a request for all attributes of the object(s) to be returned in a predefined order.		

<sup>5</sup> Material Movement Management used only the Host to Equipment direction for this message. However, both directions are included for future compatibility with Recipe Management and other future services.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S1,F20 Attribute Data (AD)	M,H<—>E
<i>Description</i>	
This message is used to transfer the requested set of object attributes. The order of requested objects and attributes is retained from the primary message.	
<i>Structure</i>	
<pre> L, 2   1. L, m                                [m = number of objects for which data is sent]       1. L, n                            [n = number of attributes returned for OBJID<sub>1</sub>]           1. &lt;ATTRDATA<sub>1</sub>&gt;           .           .           n. &lt;ATTRDATA<sub>n</sub>&gt;           .           .       m. L, n                            [n = number of attributes returned for OBJID<sub>m</sub>]           1. &lt;ATTRDATA<sub>1</sub>&gt;           .           .           n. &lt;ATTRDATA<sub>n</sub>&gt;   2. L, p                                [p = # errors reported]       1. L, 2           1. &lt;ERRCODE<sub>1</sub>&gt;           2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .       p. L, 2           1. &lt;ERRCODE<sub>p</sub>&gt;           2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If m = 0, it indicates that the specified OBJTYPE is unknown. If any n = 0, it indicates that the corresponding object was not found. If any ATTRDATA item is reported as a zero-length item, it indicates that the specified attribute does not exist. If no errors were found, p = 0.	

10.6 *Stream 2 Equipment Control and Diagnostics* — Messages which deal with control of the equipment from the host. This includes all remote operations and equipment self-diagnostics and calibration but specifically excludes the control operations which are associated with material transfer (see Stream 4), loading of executive and boot programs (Stream 8), and all file and operating system calls (Streams 10, 13). See also continuations in Stream 17.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F0 Abort Transaction (S2F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0	
<i>Structure</i>	
<i>Exception</i>	





<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F1 Service Program Load Inquire (SPI)	S,H<->E,reply
<i>Description</i>	
Either the host or equipment wants to send the specified program.	
<i>Structure</i>	
L, 2 1. <SPID> 2. <LENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F2 Service Program Load Grant (SPG)	S,H<->E
<i>Description</i>	
Provides permission to load.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F3 Service Program Send (SPS)	M,H<->E, reply
<i>Description</i>	
The data associated with the S2,F1 inquire is sent. If S2,F3 is multi-block, it must be preceded by the S2,F1/S2,F2 Inquire/Grant transaction.	
<i>Structure</i>	
<SPD>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F4 Service Program Send Acknowledge (SPA)	S,H<->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<SPAACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F5 Service Program Load Request (SPR)	S,H<->E,reply
<i>Description</i>	
A service program is requested.	
<i>Structure</i>	
<SPID>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F6 Service Program Load Data (SPD)	M,H<->E
<i>Description</i>	
A service program is sent.	
<i>Structure</i>	
<SPD>	
<i>Exception</i>	
A zero-length item means that the requested program cannot be returned.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F7 Service Program Run Send (CSS)	S,H->E,reply
<i>Description</i>	
Start the requested program.	
<i>Structure</i>	
<SPID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F8 Service Program Run Acknowledge (CSA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<CSAACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F9 Service Program Results Request (SRR)	S,H->E,reply
<i>Description</i>	
Ask for results of service program.	
<i>Structure</i>	
<SPID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F10 Service Program Results Data (SRD)	M,H<-E
<i>Description</i>	
Get the results back.	
<i>Structure</i>	
<SPR>	
<i>Exception</i>	
A zero-length item means SPR does not exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F11 Service Program Directory Request (SDR)	S,H<->E,reply
<i>Description</i>	
There may be more than one service program.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F12 Service Program Directory Data (SDD)	S,H<->E
<i>Description</i>	
A list of service program names.	
<i>Structure</i>	
L,n 1. <SPID <sub>1</sub> > . . n. <SPID <sub>n</sub> >	
<i>Exception</i>	
If n = 0, there are no service programs.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F13 Equipment Constant Request (ECR)	S,H->E,reply
<i>Description</i>	
Constants such as for calibration, servo gain, alarm limits, data collection mode, and other values that are changed infrequently can be obtained using this message.	
<i>Structure</i>	
The following structure is approved for all item formats and should be used by all new implementations:  L,n 1. <ECID <sub>1</sub> > . . n. <ECID <sub>n</sub> >	
The following structure is included for compatibility with previous implementations and may only be used for items of format 3() and 5(): <ECID <sub>1</sub> , . . . , ECID <sub>n</sub> >	
<i>Exception</i>	
A zero-length list (structure1) or item (structure2) means report all ECVs according to a predefined order.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F14 Equipment Constant Data (ECD)	M,H<-E
<i>Description</i>	
Data Response to S2,F13 in the order requested.	
<i>Structure</i>	
L, n 1. <ECV <sub>1</sub> > 2. <ECV <sub>2</sub> > . . n. <ECV <sub>n</sub> >	
<i>Exception</i>	
A zero-length list item for ECV <sub>i</sub> means that ECID <sub>i</sub> does not exist. The list format for this data item is not allowed, except in this case.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F15 New Equipment Constant Send (ECS)	S,H->E,reply
<i>Description</i>	
Change one or more equipment constants.	
<i>Structure</i>	
L, n 1. L, 2 1. <ECID <sub>1</sub> > 2. <ECV <sub>1</sub> > 2. L, 2 . . n. L, 2 1. <ECID <sub>n</sub> > 2. <ECV <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F16 New Equipment Constant Acknowledge (ECA)	S,H<-E
<i>Description</i>	
Acknowledge or error if EAC contains a non-zero error code, the equipment should not change any of the ECIDs specified in S2F15.	
<i>Structure</i>	
<EAC>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F17 Date and Time Request (DTR)	S,H<->E,reply
<i>Description</i>	
Useful to check equipment time base or for equipment to synchronize with the host time base.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F18 Date and Time Data (DTD)	S,H<->E
<i>Description</i>	
Actual time data.	
<i>Structure</i>	
<TIME>	
<i>Exception</i>	
A zero-length item means no time exists.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F19 Reset/Initialize Send (RIS)	S,H->E,reply
<i>Description</i>	
Causes equipment to reach one of several predetermined initialized conditions.	
<i>Structure</i>	
<RIC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F20 Reset Acknowledge (RIA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<RAC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F21 Remote Command Send (RCS)	S,H->E,[reply]
<i>Description</i>	
Similar to pressing buttons on the front panel or causes some equipment activity to commence or to cease.	
<i>Structure</i>	
<RCMD>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F22 Remote Command Acknowledge (RCA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<CMDA>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F23 Trace Initialize Send (TIS)	M,H->E,reply
<i>Description</i>	
<p>Status variables exist at all times. This function provides a way to sample a subset of those status variables as a function of time. The trace data is returned on S6,F1 and is related to the original request by the TRID Multiple trace requests may be made to that equipment allowing it. If equipment receives S2,F23 with the same TRID as a trace function that is currently in progress, the equipment should terminate the old trace and then initiate the new trace. A trace function currently in progress may be terminated by S2,F23 with TRID of that trace and TOTSMP = 0.</p> <p>If S2,F23 is multi-block, it must be preceded by the S2,F39/S2,F40 Inquire/Grant transaction. Some equipment may support only single-Block S6,F1, and may refuse a S2,F23 message which would cause a multi-block S6,F1.</p> <p>Each equipment shall document its trace performance limits. The Host Computer shall not send an S2,F23 which exceeds the equipment's performance limits, or the equipment may operate incorrectly.</p>	
<i>Structure</i>	
<p>The following structure is approved for all item formats and should be used by all new implementations:</p> <pre> L, 5 1. &lt;TRID&gt; 2. &lt;DSPER&gt; 3. &lt;TOTSMP&gt; 4. &lt;REPGSZ&gt; 5. L, n    1. &lt;SVID<sub>1</sub>&gt;    .    .    n. &lt;SVID<sub>n</sub>&gt; </pre> <p>The following structure is included for compatibility with previous implementations and may only be used for items whose SVID is format 3() and 5():</p> <pre> L, 5 1. &lt;TRID&gt; 2. &lt;DSPER&gt; 3. &lt;TOTSMP&gt; 4. &lt;REPGSZ&gt; 5. &lt;SVID<sub>1</sub>, . . . , SVID<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F24 Trace Initialize Acknowledge (TIA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<TIAACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F25 Loopback Diagnostic Request (LDR)	S,H<->E,reply
<i>Description</i>	
A diagnostic message for checkout of protocol and communication circuits. The binary string sent is echoed back.	
<i>Structure</i>	
<ABS>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F26 Loopback Diagnostic Data (LDD)	S,H<->E
<i>Description</i>	
The echoed binary string.	
<i>Structure</i>	
<ABS>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F27 Initiate Processing Request (IPR)	S,H->E,reply
<i>Description</i>	
Host requests equipment to initiate processing of the identified material at the specified location in the machine using the specified process program.	
<i>Structure</i>	
L, 3 1. <LOC> 2. <PPID> 3. L, n 1. <MID <sub>1</sub> > . . n. <MID <sub>n</sub> >	
<i>Exception</i>	
A zero-length PPID indicates no process program is being specified and the equipment is to take whatever action is appropriate for it to determine the proper program to use. A zero-length MID list indicates no MID is to be associated with the material to be processed.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F28 Initiate Processing Acknowledge (IPA)	S,H<-E
<i>Description</i>	
Response by equipment to Initiate Processing Request. Returned status indicates whether or not the request was honored by the equipment.	
<i>Structure</i>	
<CMDA>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F29 Equipment Constant Namelist Request (ECNR)	S,H->E,reply
<i>Description</i>	
This function allows the host to retrieve basic information about what equipment constants are available in the equipment.	
<i>Structure</i>	
L, n 1. <ECID <sub>1</sub> > . . n. <ECID <sub>n</sub> >	
<i>Exception</i>	
A zero-length list means send information for all ECIDs.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F30 Equipment Constant Namelist (ECN)	M,H<-E
<i>Description</i>	
Data Response.	
<i>Structure</i>	
L, n (number of equipment constants) 1. L, 6 1. <ECID <sub>1</sub> > 2. <ECNAME <sub>1</sub> > 3. <ECMIN <sub>1</sub> > 4. <ECMAX <sub>1</sub> > 5. <ECDEF <sub>1</sub> > 6. <UNITS <sub>1</sub> > 2. L, 6 . . n. L, 6 1. <ECID <sub>n</sub> > 2. <ECNAME <sub>n</sub> > 3. <ECMIN <sub>n</sub> > 4. <ECMAX <sub>n</sub> > 5. <ECDEF <sub>n</sub> > 6. <UNITS <sub>n</sub> >	
<i>Exception</i>	
Zero-length ASCII items for ECNAME <sub>i</sub> , ECMIN <sub>i</sub> , ECMAX <sub>i</sub> , ECDEF <sub>i</sub> , and UNITS <sub>i</sub> indicates that the ECID does not exist.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F31 Date and Time Set Request (DTS)	S,H->E,reply
<i>Description</i>	
Useful to synchronize the equipment time with the host time base.	
<i>Structure</i>	
<TIME>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F32 Date and Time Set Acknowledge (DTA)	S,H<-E
<i>Description</i>	
Acknowledge the receipt of time and date.	
<i>Structure</i>	
<TIACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F33 Define Report (DR)	M,H->E,reply
<i>Description</i>	
<p>The purpose of this message is for the host to define a group of reports for the equipment.</p> <p>The type of report to be transmitted is designated by a Boolean “Equipment Constant”. An “Equipment Constant Value” of “False” means that an “Event Report” (S6,F11) will be sent, and a value of “True” means that an “Annotated Event Report” (S6,F13) will be sent. If S2,F33 is Multi-block, it must be preceded by the S2,F39/S2,F40 Inquire/Grant transaction.</p>	
<i>Structure</i>	
<pre> L, 2   1. &lt;DATAID&gt;   2. L, a                                # reports       1. L, 2                            report 1           1. &lt;RPTID<sub>1</sub>&gt;           2. L, b                        # VIDs this report               1. &lt;VID<sub>1</sub>&gt;               .               .               b. &lt;VID<sub>b</sub>&gt;           a. L, 2                        report a               1. &lt;RPTID<sub>a</sub>&gt;               2. L, c                    # VIDs this report                   1. &lt;VID<sub>1</sub>&gt;                   .                   .                   c. &lt;VID<sub>c</sub>&gt; </pre>	
<i>Exception</i>	
<p>1. A list of zero-length following &lt;DATAID&gt; deletes all report definitions and associated links. See S2,F35 (Link Event/Report).</p> <p>2. A list of zero-length following &lt;RPTID&gt; deletes report type RPTID. All CEID links to this RPTID are also deleted.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F34 Define Report Acknowledge (DRA)	S,H<-E
<i>Description</i>	
Acknowledge or error if an error condition is detected the entire message is rejected (i.e., partial changes are not allowed).	
<i>Structure</i>	
<DRACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F35 Link Event Report (LER)	M,H->E,reply
<i>Description</i>	
<p>The purpose of this message is for the host to link n reports to an event (CEID). These linked event reports will default to 'disabled' upon linking. That is, the occurrence of an event would not cause the report to be sent until enabled. See S2,F37 for enabling reports.</p> <p>If S2,F35 is Multi-block, it must be preceded by the S2,F39/S2,F40 Inquire/Grant transaction.</p>	
<i>Structure</i>	
<pre> L, 2   1. &lt;DATAID&gt;   2. L, a                                # events       1. L, 2                            event 1           1. &lt;CEID<sub>1</sub>&gt;           2. L, b               1. &lt;RPTID<sub>1</sub>&gt;               .               .               b. &lt;RPTID<sub>b</sub>&gt;           .           .       a. L, 2                            event a           1. &lt;CEID<sub>a</sub>&gt;                    # RPTIDS this event           2. L, c               1. &lt;RPTID<sub>1</sub>&gt;               .               .               c. &lt;RPTID<sub>c</sub>&gt; </pre>	
<i>Exception</i>	
A list of zero length following CEID deletes all report links to that event.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F36 Link Event Report Acknowledge (LERA)	S,H<-E
<i>Description</i>	
Acknowledge or error If an error condition is detected the entire message is rejected (i.e., partial changes are not allowed).	
<i>Structure</i>	
<LRACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F37 Enable/Disable Event Report (EDER)	S,H->E,reply
<i>Description</i>	
The purpose of this message is for the host to enable or disable reporting for a group of events (CEIDs).	
<i>Structure</i>	
L, 2 1. <CEED> enable/disable 2. L, n #CEIDs 1. <CEID <sub>1</sub> > . . n. <CEID <sub>n</sub> >	
<i>Exception</i>	
A list of zero length following <CEED> means all CEIDs.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F38 Enable/Disable Event Report Acknowledge (EERA)	S,H<-E
<i>Description</i>	
Acknowledge or error if an error condition is detected the entire message is rejected, i.e., partial changes are not allowed.	
<i>Structure</i>	
<ERACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F39 Multi-block Inquire (DMBI)	S,H->E,reply
<i>Description</i>	
If a S2,F23, S2,F33, S2,F35, S2,F45, or S2,F49 message is more than one block, this transaction must precede the message.	
<i>Structure</i>	
L, 2 1. <DATAID> 2. <DATALENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F40 Multi-block Grant (DMBG)	S,H<-E
<i>Description</i>	
Grant permission to send multi-block message.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F41 Host Command Send (HCS)	S,H->E,reply
<i>Description</i>	
The Host requests the Equipment perform the specified remote command with the associated parameters.	
<i>Structure</i>	
L, 2 1. <RCMD> 2. L, n # of parameters 1. L, 2 1. <CPNAME <sub>1</sub> > parameter 1 name 2. <CPVAL <sub>1</sub> > parameter 1 value . . n. L, 2 1. <CPNAME <sub>n</sub> > parameter n name 2. <CPVAL <sub>n</sub> > parameter n value	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F42 Host Command Acknowledge (HCA)	S,H<-E
<i>Description</i>	
Acknowledge Host command or error. If command is not accepted due to one or more invalid parameters (i.e., HCAACK = 3), then a list of invalid parameters will be returned containing the parameter name and reason for being invalid.	
<i>Structure</i>	
L, 2 1. <HCAACK> 2. L, n # of parameters 1. L, 2 1. <CPNAME <sub>1</sub> > parameter 1 name 2. <CPACK <sub>1</sub> > parameter 1 reason . . n. L, 2 1. <CPNAME <sub>n</sub> > parameter n name 2. <CPACK <sub>n</sub> > parameter n reason	
<i>Exception</i>	
If there are no invalid parameters, then a list of zero length will be sent for item 2.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F43 Reset Spooling Streams and Functions (RSSF)	S,H->E,reply
<i>Description</i>	
This message allows the host to select specific streams and functions to be spooled whenever spooling is active.	
<i>Structure</i>	
<pre> L, m   1. L, 2     1. &lt;STRID<sub>1</sub>&gt;     2. L, n       1. &lt;FCNID<sub>1</sub>&gt;       .       .       n. &lt;FCNID<sub>n</sub>&gt;     .     .   m. L, 2     1. &lt;STRID<sub>m</sub>&gt;     2. L, n       1. &lt;FCNID<sub>1</sub>&gt;       .       .       n. &lt;FCNID<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
<ol style="list-style-type: none"> <li>1. A zero-length list, m = 0, turns off spooling for all streams and functions.</li> <li>2. A zero-length list, n = 0, turns on spooling for all functions for the associated stream.</li> </ol> <p>Notes:</p> <ol style="list-style-type: none"> <li>1. Turning off spooling for all functions for a specific stream is achieved by omitting reference to the stream from this message.</li> <li>2. Spooling for Stream 1 is not allowed.</li> <li>3. Equipment must allow host to spool all primary messages for a stream (except Stream 1).</li> <li>4. A defined list of functions for a stream in this message will replace any previously selected functions.</li> </ol>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F44 Reset Spooling Acknowledge (RSA)	M,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<pre> L, 2   1. &lt;RSPACK&gt;                                (accept or reject)   2. L, m                                       (m = number of streams with errors)       1. L, 3           1. &lt;STRID<sub>1</sub>&gt;           2. &lt;STRACK<sub>1</sub>&gt;                        (error in stream)           3. L, n                             (n = number of functions in error)               1. &lt;FCNID<sub>1</sub>&gt;               .               .               n. &lt;FCNID<sub>n</sub>&gt;           .           .       m. L, 3           1. &lt;STRID<sub>m</sub>&gt;           2. &lt;STRACK<sub>m</sub>&gt;                        (error in stream)           3. L, n                             (n = number of functions in error)               1. &lt;FCNID<sub>1</sub>&gt;               .               .               n. &lt;FCNID<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
1. If RSPACK = 0, a zero-length list, m = 0, is given, indicating no streams or functions in error. 2. A zero-length list, n = 0, indicates no functions in error for specified stream.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F45 Define Variable Limit Attributes (DVLA)	M,H->E,reply
<i>Description</i>	
<i>Structure</i>	
<pre> L, 2     1. &lt;DATAID&gt;     2. L,m                (m = # of variables in this definition)         1. L,2             1. &lt;VID<sub>1</sub>&gt;             2. L,n        (n = # of limits being defined/changed for VID<sub>1</sub>)                 1. L,2                     1. &lt;LIMITID<sub>1</sub>&gt;                     2. L,p        (p = {0,2})                         1. &lt;UPPERDB<sub>1</sub>&gt;                         2. &lt;LOWERDB<sub>1</sub>&gt;                         .                         .                     n. L,2                         1. &lt;LIMITID<sub>n</sub>&gt;                         2. L,p        (p = {0,2})                             1. &lt;UPPERDB<sub>n</sub>&gt;                             2. &lt;LOWERDB<sub>n</sub>&gt;                 .             .         m.L, 2             1. &lt;VID<sub>m</sub>&gt;             2. L,n        (n = # of limits being defined/changed for VID<sub>m</sub>)                 1. L,2                     1. &lt;LIMITID<sub>1</sub>&gt;                     2. L,p        (p = {0,2})                         1. &lt;UPPERDB<sub>1</sub>&gt;                         2. &lt;LOWERDB<sub>1</sub>&gt;                 .                 .             n. L,2                 1. &lt;LIMITID<sub>n</sub>&gt;                 2. L,p        (p = {0,2})                     1. &lt;UPPERDB<sub>n</sub>&gt;                     2. &lt;LOWERDB<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
<ol style="list-style-type: none"> <li>1. A zero-length list, m = 0, sets all limit values for all monitored VIDs to “undefined”.</li> <li>2. A zero-length list, n = 0, sets all limits values for that VID to “undefined”.</li> <li>3. A zero-length list, p = 0, sets that limit to “undefined”.</li> </ol>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F46 Variable Limit Attribute Acknowledge (VLAA)	M,H<-E
<i>Description</i>	
Acknowledge definition of variable limit attributes or report error. If DVLA is not accepted due to one or more invalid parameters (e.g., LIMITACK = 3), then a list of invalid parameters is returned containing the variable limit attribute and reason for rejection. If an error condition is detected, the entire message is rejected (i.e., partial changes are not allowed).	
<i>Structure</i>	
<pre> L, 2   1. &lt;VLAACK&gt;   2. L, m                                     (m = number of invalid parameters)       1. L, 3           1. &lt;VID<sub>1</sub>&gt;                         (VID with error)           2. &lt;LVACK<sub>p</sub>&gt;                       (reason)           3. L, n                             {n = 0, 2}               1. &lt;LIMITID<sub>1</sub>&gt;                (1st limit in error for VID<sub>p</sub>)               2. &lt;LIMITACK<sub>1</sub>&gt;              (reason)           .           .       m. L, 3           1. &lt;VID<sub>m</sub>&gt;                         (VID with error)           2. &lt;LVACK<sub>m</sub>&gt;                       (reason)           3. L, n                             {n = 0, 2}               1. &lt;LIMITID<sub>1</sub>&gt;                (1st limit in error for VID<sub>x</sub>)               2. &lt;LIMITACK<sub>1</sub>&gt;              (reason) </pre>	
<i>Exception</i>	
1. A zero-length list, m = 0 indicates no invalid variable limit attributes. 2. A zero-length list, n = 0 indicates no invalid limit values for that VID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F47 Variable Limit Attribute Request (VLAR)	S,H->E,reply
<i>Description</i>	
This message allows the host to query the equipment for current variable limit attribute definitions.	
<i>Structure</i>	
<pre> L, m                                     (m = # of VIDs this request)   1. &lt;VID<sub>1</sub>&gt;   .   .   m. &lt;VID<sub>m</sub>&gt; </pre>	
<i>Exception</i>	
A zero-length list, m = 0, requests a list of all VID values that can have variable limit attributes.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F48 Variable Limit Attributes Send (VLAS)	M,H<-E
<i>Description</i>	
Equipment sends values of requested variable limit attribute definitions in the order requested.	
<i>Structure</i>	
<pre> L,m                                     (m = # of VIDs this request)   1. L,2     1. &lt;VID<sub>1</sub>&gt;     2. L,p                             {p = 0,4}       1. &lt;UNITS<sub>1</sub>&gt;       2. &lt;LIMITMIN<sub>1</sub>&gt;       3. &lt;LIMITMAX<sub>1</sub>&gt;       4. L,n                             (n = # of limits defined for this VID)         1. L,3           1. &lt;LIMITID<sub>1</sub>&gt;           2. &lt;UPPERDB<sub>1</sub>&gt;           3. &lt;LOWERDB<sub>1</sub>&gt;           .           .         n. L,3           1. &lt;LIMITID<sub>n</sub>&gt;           2. &lt;UPPERDB<sub>n</sub>&gt;           3. &lt;LOWERDB<sub>n</sub>&gt;       .       . m. L,2   1.&lt;VID<sub>m</sub>&gt;   2. L,p                             {p = 0,4}     1. &lt;UNITS<sub>m</sub>&gt;     2. &lt;LIMITMIN<sub>m</sub>&gt;     3. &lt;LIMITMAX<sub>m</sub>&gt;     4. L,n                             (n = # of limits defined for this VID)       1. L,3         1. &lt;LIMITID<sub>1</sub>&gt;         2. &lt;UPPERDB<sub>1</sub>&gt;         3. &lt;LOWERDB<sub>1</sub>&gt;         .         .       n. L,3         1. &lt;LIMITID<sub>n</sub>&gt;         2. &lt;UPPERDB<sub>n</sub>&gt;         3. &lt;LOWERDB<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
1. A zero-length list, p = 0, indicates that limits are not supported for the VID. 2. A zero-length list, n = 0, means no limits are currently defined for the specified variable.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F49 Enhanced Remote Command	M,H->E
<b>Description</b>	
The host requests an object to perform the specified remote command with its associated parameters. If multi-block, it shall be preceded by the S2,F39/S2,F40 Multi-Block Inquire/Grant transaction.	
<b>Structure</b>	
L, 4 1. <DATAID> 2. <OBJSPEC> 3. <RCMD> 4. L,m # of parameter groups 1. L,2 1. <CPNAME <sub>1</sub> > command parameter 1 name 2. <CEPVAL <sub>1</sub> > command-enhanced parameter 1 value 2. L,2 1. <CPNAME <sub>2</sub> > command parameter 2 name 2. <CEPVAL <sub>2</sub> > command-enhanced parameter 2 value . . m. L,2 1. <CPNAME <sub>m</sub> > command parameter m name 2. <CEPVAL <sub>m</sub> > command enhanced parameter m value	
If a specific value of CPNAME is defined to have a CEPVAL defined as a LIST, it shall always be a LIST. If the CEPVAL that is associated to that specific value of CPNAME is defined to be anything other than LIST, it will result in a format error.	
<b>Exception</b>	
A zero length list, m = 0, indicates that no parameter groups are sent with the command. OBJSPEC can be a null length item.	
Notes:	
1. If CEPVAL is a LIST, the items that make up that list shall take on one of the following forms: (1) a list of items with an identical format, (2) a LIST of CPNAME, CEPVAL pairs, as illustrated below.	
<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> A) L, 2  1. &lt;CPNAME<sub>a</sub>&gt;  2. L,m      1. &lt;CPVAL<sub>a1</sub>&gt;      2. &lt;CPVAL<sub>a2</sub>&gt;      .      .      m. &lt;CPVAL<sub>am</sub>&gt; </div> <div style="width: 45%;"> B) L, 2  1. &lt;CPNAME<sub>b</sub>&gt;  2. L,n      1. L, 2          1. &lt;CPNAME<sub>b1</sub>&gt;          2. &lt;CEPVAL<sub>b1</sub>&gt;      .      .      n. L, 2          1. &lt;CPNAME<sub>bn</sub>&gt;          2. &lt;CEPVAL<sub>bn</sub>&gt; </div> </div>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S2,F50 Enhanced Remote Command Acknowledge	M,H<-E
<i>Description</i>	
The equipment acknowledges Enhanced Remote Command or reports any error(s). If the command is not accepted due to one or more invalid parameters, (i.e., HCACK = 3), then a list of invalid parameters will be returned containing the parameter name and reason for being invalid.	
<i>Structure</i>	
L, 2 1. <HCACK> 2. L, n # of parameter groups 1. L, 2 1. <CPNAME <sub>1</sub> > 2. <CEPACK <sub>1</sub> > . . n. L, 2 1. <CPNAME <sub>n</sub> > 2. <CEPACK <sub>n</sub> >	
<i>Exception</i>	
None	

10.7 *Stream 3 Materials Status* — The functions of the material status stream are used to communicate information and actions related to material, including carriers and material-in-process, time-to-completion information, and extraordinary material occurrences.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F0 Abort Transaction (S3F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F1 Material Status Request (MSR)	S,H->E,reply
<i>Description</i>	
Host requests the device to send the status of all material in process.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F2 Material Status Data (MSD)	M,H<-E
<i>Description</i>	
Material-in-process information is sent from the equipment to the host. There are m locations.	
<i>Structure</i>	
L, 2 <ul style="list-style-type: none"> <li>1. &lt;MF&gt;</li> <li>2. L, m <ul style="list-style-type: none"> <li>1. L, 3 <ul style="list-style-type: none"> <li>1. &lt;LOC<sub>1</sub>&gt;</li> <li>2. &lt;QUA<sub>1</sub>&gt;</li> <li>3. &lt;MID<sub>1</sub>&gt;</li> </ul> </li> <li>2. L, 3</li> <li>.</li> <li>.</li> </ul> </li> <li>m. L, 3 <ul style="list-style-type: none"> <li>1. &lt;LOC<sub>m</sub>&gt;</li> <li>2. &lt;QUA<sub>m</sub>&gt;</li> <li>3. &lt;MID<sub>m</sub>&gt;</li> </ul> </li> </ul>	
<i>Exception</i>	
A zero-length list returned means no such data exists.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F3 Time to Completion Request (TCR)	S,H->E,reply
<i>Description</i>	
Host requests the equipment to send the time-to-completion of operations on all material in possession.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F4 Time to Completion Data (TCD)	M,H<-E
<i>Description</i>	
Time-to-completion information is sent by the equipment to the host.	
<i>Structure</i>	
L, 2 <ul style="list-style-type: none"> <li>1. &lt;MF&gt;</li> <li>2. L, m <ul style="list-style-type: none"> <li>1. L, 3 <ul style="list-style-type: none"> <li>1. &lt;TTC<sub>1</sub>&gt;</li> <li>2. &lt;QUA<sub>1</sub>&gt;</li> <li>3. &lt;MID<sub>1</sub>&gt;</li> </ul> </li> <li>2. L, 3</li> <li>.</li> <li>.</li> </ul> </li> <li>m. L, 3 <ul style="list-style-type: none"> <li>1. &lt;TTC<sub>m</sub>&gt;</li> <li>2. &lt;QUA<sub>m</sub>&gt;</li> <li>3. &lt;MID<sub>m</sub>&gt;</li> </ul> </li> </ul>	



<i>Exception</i>
A zero-length list header returned means no such data exists.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F5 Material Found Send (MFS)	S,H<-E,[reply]
<i>Description</i>	
The equipment advises the host that unsolicited material has appeared at one of its sensors.	
<i>Structure</i>	
L, 2	
1. <MF>	
2. <QUA>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F6 Material Found Acknowledge (MFA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC3>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F7 Material Lost Send (MLS)	S,H<-E,[reply]
<i>Description</i>	
The equipment advises the host that material has disappeared from its sensors.	
<i>Structure</i>	
L, 3	
1. <MF>	
2. <QUA>	
3. <MID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F8 Material Lost Acknowledge (MLA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC3>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F9 Material ID Equate Send (IES)	S,H<-E,reply
<i>Description</i>	
Provide an alternative name to be used as equivalent to the original material ID.	
<i>Structure</i>	
L, 2 1. <MID> 2. <EMID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F10 Material ID Equate Acknowledge (IEA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC3>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F11 Material ID Request (MIDR)	S,H<-E,reply
<i>Description</i>	
The equipment requests the Material ID of the material at the specified port.	
<i>Structure</i>	
<PTN>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F12 Material ID Request Acknowledge (MIRA)	S,H->E
<i>Description</i>	
The host acknowledges the request for the Material ID. If the use of a request/acknowledge/send/acknowledge conversation is required, it indicated by the acknowledge code MIDRA = 2. In this case, the send/acknowledge transaction is S3,F13, S3,F14. A timeout when electing S3,F13 is indicated by S9,F13 or a restart of the conversation, with S3,F11.	
<i>Structure</i>	
L, 3 1. <PTN> 2. <MIDRA> 3. <MID>	
Note: For all cases except MIDRA = 0 (accepted, <MID> follows), the <MID> will be ignored by the receiver of message S3,F12. When MIDRA = 0, a zero-length MID indicates that no MID is available.	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F13 Material ID Send (MIS)	S,H->E,reply
<i>Description</i>	
The host sends the Material ID of the material at the specified port.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
Note: A zero-length MID indicates that no MID is available.	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F14 Material ID Acknowledge (MIA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<MIDAC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F15 Materials Multi-Block Inquire (MMBI)	S,H->E,reply
<i>Description</i>	
This message requests permission to send a multi-block message based upon a maximum length of the total message. It must be sent prior to sending any multi-block primary message in Stream 3.	
<i>Structure</i>	
L, 2 1. <DATAID> 2. <DATALENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F16 Materials Multi-Block Grant (MMBG)	S,H<-E
<i>Description</i>	
This message grants or denies permission to send a multi-block primary message in Stream 3.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F17 Carrier Action Request	M,H->E,reply
<i>Description</i>	
This message requests an action to be performed for a specified carrier. If multi-block, this message must be preceded by the S3,F15/F16 transaction.	
<i>Structure</i>	
<pre> L, 5   1. &lt;DATAID&gt;   2. &lt;CARRIERACTION&gt;   3. &lt;CARRIERID&gt;   4. &lt;PTN&gt;   5. L, n                                n = number of carrier attributes       1. L, 2           1. &lt;CATTRID<sub>1</sub>&gt;           2. &lt;CATTRDATA<sub>1</sub>&gt;       .       .       n. L, 2           1. &lt;CATTRID<sub>n</sub>&gt;           2. &lt;CATTRDATA<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
<p>If n = 0, then no carrier attributes are included. If CARRIERID is not a zero-length item, then PTN may be omitted (a zero-length item). ATTRID and ATTRDATA may be substituted for CATTRID and CATTRDATA respectively. ReticlePodLocationID may be used as one of &lt;CATTRID&gt; when the CARRIERACTION is PodRelease and the carrier is not at a Load Port.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F18 Carrier Action Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the carrier action request.	
<i>Structure</i>	
<pre> L, 2   1. &lt;CAACK&gt;   2. L, n       1. L, 2           1. &lt;ERRCODE<sub>1</sub>&gt;           2. &lt;ERRTEXT<sub>1</sub>&gt;       .       .       n. L, 2           1. &lt;ERRCODE<sub>n</sub>&gt;           2. &lt;ERRTEXT<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, no errors exist.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F19 Cancel All Carrier Out Request	S,H->E, reply
<i>Description</i>	
This message is used to cancel all pending carrier out requests.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F20 Cancel All Carrier Out Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the Cancel Carrier Out request.	
<i>Structure</i>	
L, 2 1. <CAACK> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F21 Port Group Definition	S,H->E, reply
<i>Description</i>	
This message defines the port in a port group and provides the initial port access.	
<i>Structure</i>	
L, 3 1. <PORTGRPNAME> 2. <PORTACCESS> 3. L, n 1. <PTN <sub>1</sub> > . . n. <PTN <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F22 Port Group Definition Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the port group definition.	
<i>Structure</i>	
L, 2 1. <CAACK> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F23 Port Group Action Request	S,H->E, reply
<i>Description</i>	
This message requests an action be performed for a port group. The access mode may be changed or the port group may be deleted.	
<i>Structure</i>	
L, 3 1. <PGRP ACTION> 2. <PORTGRPNAME> 3. L, m 1. L, 2 1. <PARAMNAME <sub>1</sub> > 2. <PARAMVAL <sub>1</sub> > . . m. L, 2 1. <PARAMNAME <sub>m</sub> > 2. <PARAMVAL <sub>m</sub> >	
<i>Exception</i>	
If m = 0, then no parameters are provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F24 Port Group Action Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the port group action.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;CAACK&gt;</li> <li>2. L, n <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>n</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F25 Port Action Request	S,H->E, reply
<i>Description</i>	
This message requests an action be performed for a port.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;PORTACTION&gt;</li> <li>2. &lt;PTN&gt;</li> <li>3. L, m <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;PARAMNAME<sub>1</sub>&gt;</li> <li>2. &lt;PARAMVAL<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>m. L, 2 <ol style="list-style-type: none"> <li>1. &lt;PARAMNAME<sub>m</sub>&gt;</li> <li>2. &lt;PARAMVAL<sub>m</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
If m = 0, then no parameters are provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F26 Port Action Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the port action request.	
<i>Structure</i>	
L, 2 1. <CAACK> 2. L, n 1. L, 2 3. <ERRCODE <sub>1</sub> > 4. <ERRTEXT <sub>1</sub> > . . n. L, 2 5. <ERRCODE <sub>n</sub> > 6. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F27 Change Access	S,H->E, reply
<i>Description</i>	
The Host requests the Equipment to change the Access Mode for the specified Load Ports. ACCESSMODE specifies the desired Access Mode. PTN specifies a desired Load Port Number.	
<i>Structure</i>	
L, 2 1. <ACCESSMODE> 2. L, n 1. <PTN <sub>1</sub> > . . n. <PTN <sub>n</sub> >	
<i>Exception</i>	
If n = 0, then the command applies to all Load Ports on the equipment. If any specified port is already in the specified Access Mode, then the Equipment shall accept the command, and toggle all loadports to specified mode. If the Equipment is unable to change one or more of specified Port(s) to the specified Access Mode, then the Equipment shall accept the command (with appropriate response acknowledgement), and shall change only the Access Mode of those Port(s) allowed by the equipment, supplying the host with an indication that not all ports were successfully changed.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F28 Change Access Acknowledge	S,H<-E
<i>Description</i>	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;CAACK&gt;</li> <li>2. L, n <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;PTN<sub>1</sub>&gt;</li> <li>2. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>3. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> </ol> </li> <li>n. L, 3 <ol style="list-style-type: none"> <li>1. &lt;PTN<sub>n</sub>&gt;</li> <li>2. &lt;ERRCODE<sub>n</sub>&gt;</li> <li>3. &lt;ERRTEXT<sub>n</sub>&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
If the command is successful, CAACK = 0, and n = 0. If the command was successful for some ports, CAACK = 6, and n > 0.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F29 Carrier Tag Read Request	S,H->E, reply
<i>Description</i>	
<p>The host requests the equipment to read data from the carrier tag of a carrier. The carrier must be identified either by its location identifier or its carrier identifier, or both. DATASEG may be used to indicate a specific section of data to be read. DATALENGTH is used to limit the amount of data for that section.</p>	
<i>Structure</i>	
L, 4 <ol style="list-style-type: none"> <li>1. &lt;LOCID&gt;</li> <li>2. &lt;CARRIERSPEC&gt;</li> <li>3. &lt;DATASEG&gt;</li> <li>4. &lt;DATALENGTH&gt;</li> </ol>	
<i>Exception</i>	
<p>Either LOCID and CARRIERSPEC can omitted (zero length item), but not both. If DATASEG and DATALENGTH are both omitted (are zero length items) then all data is requested. If DATALENGTH only is omitted, then all data within the indicated section is requested.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F30 Carrier Tag Read Data (CTRD)	S,H<-E
<i>Description</i>	
This message is used to return requested information from the carrier tag of the carrier indicated in the request and to acknowledge the results of the request.	
<i>Structure</i>	
L, 2 1. <DATA> 2. <L, 2> 1. <CAACK> 2. L, s 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . s. L, 2 1. <ERRCODE <sub>s</sub> > 2. <ERRTEXT <sub>s</sub> >	
<i>Exception</i>	
If the carrier identifier or the carrier location originally specified is unknown, then DATA is zero length. If CAACK is non-zero, then DATA is zero length.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F31 Carrier Tag Write Data Request (CTWDR)	S,H->E, reply
<i>Description</i>	
The host requests the equipment to write data to a carrier tag. The carrier must be indicated either by its location identifier or its carrier identifier, or both. DATASEG may be used to indicate a specific section of data to be written or overwritten. DATALENGTH may be used to indicate the length of the data to be written.	
<i>Structure</i>	
L, 5 1. <LOCID> 2. <CARRIERSPEC> 3. <DATASEG> 4. <DATALENGTH> 5. <DATA>	
<i>Exception</i>	
Either LOCID and CARRIERSPEC can be omitted (zero length item), but not both. If DATASEG and DATALENGTH are both omitted (are zero length items) then all data is to be overwritten. If only DATALENGTH is omitted, then all data within the indicated section is to be written.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F32 Carrier Tag Write Data Acknowledge (CTWDA)	S,H<-E
<i>Description</i>	
This message acknowledges the success or failure of writing data to the carrier tag requested.	
<i>Structure</i>	
L, 2 1. <CAACK> 2. L, s 1. L, 2 1. <ERRORCODE <sub>1</sub> > 2. <ERRORTTEXT <sub>1</sub> > . . s. L, 2 1. <ERRORCODE <sub>s</sub> > 2. <ERRORTTEXT <sub>s</sub> >	
<i>Exception</i>	
s = 0 if and only if there are no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3, F33 Cancel All Pod Out Request	
<i>Description</i>	
This message is used to cancel all pending pod out requests.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F34 Cancel All Pod Out Acknowledge	
<i>Description</i>	
This message acknowledges the Cancel Pod Out request.	
<i>Structure</i>	
L, 2 1. <CAACK> 2. L, n 1. L, 2 1. <ERRORCODE <sub>1</sub> > 2. <ERRORTTEXT <sub>1</sub> > . . n. L, 2 1. <ERRORCODE <sub>n</sub> > 2. <ERRORTTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F35 Reticle Transfer Job Request	
<i>Description</i>	
This message requests a reticle transfer job be performed (or cancelled) for a particular pod.	
<i>Structure</i>	
<pre> L, 6   1. &lt;JOBACTION&gt;   2. &lt;PODID&gt;   3. &lt;INPTN&gt;   4. &lt;OUTPTN&gt;   5. L, n                                n = number of attributes       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       .       n. L, 2           1. &lt;ATTRID<sub>n</sub>&gt;           2. &lt;ATTRDATA<sub>n</sub>&gt;   5. L, m                                m = capacity       1. L, 3           1. &lt;RETICLEID<sub>1</sub>&gt;           2. &lt;RETREMOVEINSTR<sub>1</sub>&gt;           3. L, r                        r = number of attributes               1. L, 2                   1. &lt;ATTRID<sub>1.1</sub>&gt;                   2. &lt;ATTRDATA<sub>1.1</sub>&gt;               .               .               .               r. L, 2                   1. &lt;ATTRID<sub>1.r</sub>&gt;                   2. &lt;ATTRDATA<sub>1.r</sub>&gt;           .           .           .       m. L, 3           1. &lt;RETICLEID<sub>m</sub>&gt;           2. &lt;RETREMOVEINSTR<sub>m</sub>&gt;           3. L, r                        r = number of attributes               1. L, 2                   1. &lt;ATTRID<sub>m.1</sub>&gt;                   2. &lt;ATTRDATA<sub>m.1</sub>&gt;               .               .               .               r. L, 2                   1. &lt;ATTRID<sub>m.r</sub>&gt;                   2. &lt;ATTRDATA<sub>m.r</sub>&gt;   6. L, m                                m = capacity       1. L, 2           1. &lt;RETICLEID<sub>1</sub>&gt;           2. &lt;RETPLACEINSTR<sub>1</sub>&gt;       . </pre>	



. m. L, 2 1. <RETICLEID <sub>m</sub> > 2. <RETPLACEINSTR <sub>m</sub> >
<i>Exception</i>
If JOBACTION = CancelReticleTransferJob, m and n may = 0.

<i>Stream, Function Name (Mnemonic)</i>	<i>Direction</i>
S3,F36 Reticle Transfer Job Request Acknowledgement	
<i>Description</i>	
This message acknowledges the ReticleTransferJobRequest.	
<i>Structure</i>	
L, 2 1. <RPMACK> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
None	

10.8 *Stream 4 Material Control* — The material control stream contains the original material control protocol and the newer protocol which supports SEMI E32.

10.8.1 *Original Material Control Protocol* — The functions in the material control stream are used to effect the automatic transfer of material between equipment. A simple handshake is achieved which provides for a variety of error conditions which gracefully terminate the handshake. Separate messages advise the host of errors and completed material transfers.

10.8.1.1 Since the handshake and host messages are separate, the handshake may be achieved transparently through the host or by direct connection between equipment. The host completes the handshake by relaying messages between the equipment. Only a single port is required on the equipment, and the equipment has a simple message handling requirement. When a direct connection is desired, at least three ports are required, the receiving equipment must look like the host with respect to the sending equipment, and message handling in the equipment is significantly more complicated than in the host-only connection. Nevertheless, the direct connection may still be chosen in an attempt to provide operation without a host. Since the host is reasonably transparent in the material handshake, a simple explanation of the handshake may be achieved by just considering the exchange of messages between the sender, the equipment wanting to get rid of material, and the receiver (the equipment able to accept the material).

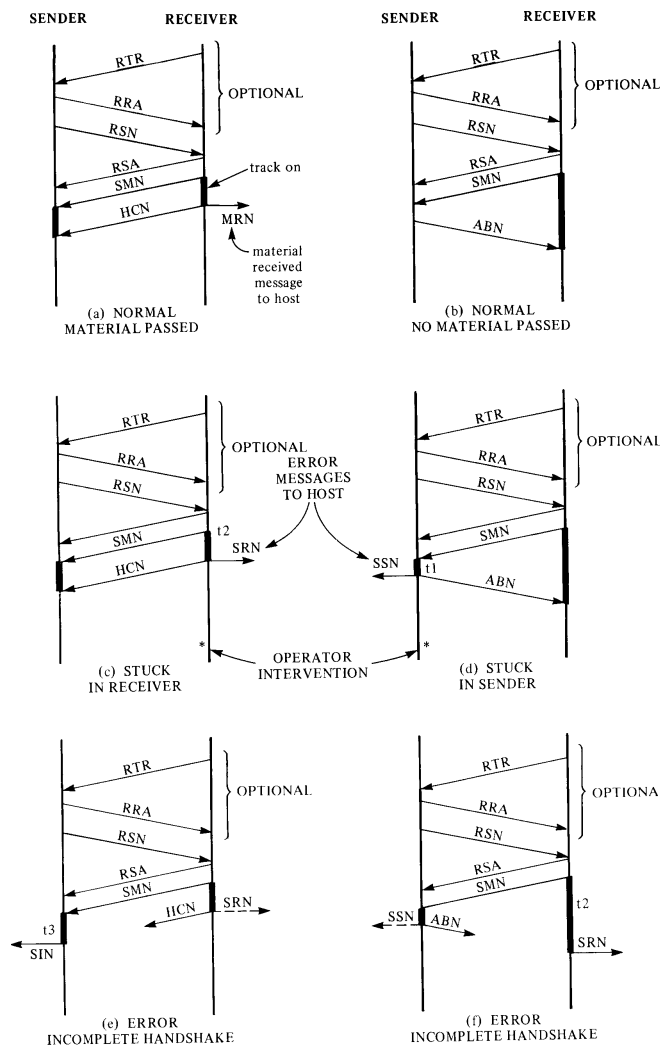
10.8.1.2 Figure 4 shows six possible handshake situations between the sender and the receiver. There are two normal handshakes. Figure 4(a) shows the normal three-message exchange when material is passed between equipment. The host is informed of a complete transfer of material. Figure 4(b) shows an alternative message exchange where the sender changes its mind and decides not to send the material. Figures 4(c) and (d) show two situations where the material gets stuck during the transfer. In each situation an error message is issued to the host from the equipment where the material is stuck. The other equipment terminates normally. When material is stuck, manual intervention is required to move the material towards the equipment which indicates the stuck condition. The manual intervention has two possible outcomes. One, the material can be moved to a position where the handshake can resume or, two, the material is broken or lost from the transfer. Lost material causes a lost material error

message to be sent to the host prior to resuming the operation. The specific details of recovering front stuck material are equipment-dependent. The stuck material condition is determined by the amount of time the material transfer mechanism is turned on. The sender claims stuck material if the material is not clear of its sensor before a time  $t_1$ . The receiver claims stuck material if the material is not received before time  $t_2$ . Figures 4(e) and (f) show the possible error conditions in the unlikely event that for some reason a handshake message is lost. Figure 4(e) shows that time  $t_3$  is the longest that the sender will wait for material received message. Times  $t_2$  and  $t_3$  set an upper limit on the amount of time either material transport mechanism will operate.

10.8.1.3 Figure 5 summarizes the interaction of the timers, handshake messages, and the error messages in the form of a flow chart. It also identifies specific states for the sender and the receiver. These states are referred to in the messages. The ranges of timer values are as follows:

- $t_1$  — time to leave sender
- $t_1 + 10 \leq t_2 \leq 60$  sec. — time to receive
- $t_2 + 10 \leq t_3 \leq 70$  sec. — time to complete send
- Default values,  $t_1 = 10$  sec.,  $t_2 = 60$  sec.,  $t_3 = 70$  sec.

NOTE 3:  $t_1$ ,  $t_2$ ,  $t_3$  defined for Stream 4 are not to be confused with timeouts T1, T2, T3, and T4 defined in SEMI E4 (SECS-I).



**Figure 4**  
**The Six Possible Handshakes**



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F0 Abort Transaction (S4F0)	S,H->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F1 Ready to Send Materials (RSN)	S,H<->E,reply
<i>Description</i>	
The sender advises the receiver that some material is awaiting transfer.	
<i>Structure</i>	
L, 2	
1. <PTN>	
2. <MID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F2 Ready to Send Acknowledge (RSA)	S,H<->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<RSACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F3 Send Material (SMN)	S,H<->E
<i>Description</i>	
The receiver advises the sender that it is ready to receive material and that its transfer mechanism is running.	
<i>Structure</i>	
L, 2	
1. <PTN>	
2. <MID>	
<i>Exception</i>	
None	

S4,F4 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F5 Handshake Complete (HCN)	S,H<->E
<i>Description</i>	
Receiver advises sender that the handshake is complete. The sender may now stop its transfer mechanism.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
None	

#### S4,F6 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F7 Not Ready to Send (ABN)	S,H<->E
<i>Description</i>	
Sender advises receiver that no material is being sent. The receiver may now stop its transfer mechanism.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
None	

#### S4,F8 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F9 Stuck in Sender (SSN)	S,H<-E
<i>Description</i>	
An error from the sender to the host. The time between the receipt of Material (SMN) and the material leaving the sender's sensor exceeds the sender's t1 timeout. The sender goes to a hold state until the disposition of the stuck material is determined.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
None	

#### S4,F10 Not Used



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F11 Stuck in Receiver (SRN)	S,H<-E
<i>Description</i>	
An error from the receiver to the host. The time between Send Material (SMN) and detection of the material at the receiver exceeds the receiver's t2 timeout. The receiver goes to a hold state until the disposition of material is determined.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
None	

S4,F12 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F13 Send Incomplete Timeout (SIN)	S,H<-E
<i>Description</i>	
An error from the sender to the host. The time between the receipt of the Send Material (SMN) and the receipt of Handshake-Complete (HCN) exceeds the sender's t3 timeout. There has been an error in the handshake and the transfer mechanism is turned off.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
None	

S4,F14 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F15 Material Received (MRN)	S,H<-E
<i>Description</i>	
A message from the receiver to the host. Material has been transferred to the receiver.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
None	

S4,F16 Not Used



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F17 Request to Receive (RTR)	S,H<->E,reply
<i>Description</i>	
Receiver requests the sender initiate a conversation to send the specified material to the specified port.	
<i>Structure</i>	
L, 2 1. <PTN> 2. <MID>	
<i>Exception</i>	
A zero-length MID means equipment doesn't know MID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F18 Request to Receive Acknowledge (RRA)	S,H<->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<RRACK>	
<i>Exception</i>	
None	

10.8.2 *Support for Material Movement Management Services* — The following messages were defined to support SEMI E32.

10.8.2.1 *Macro Level Messages* — The following messages support the host supervised macro level of material movement as defined in SEMI E32. Stream 1 Macro Level Messages can be found in § 10.5: S1F19, Get Attribute (GA); S1F20, Attribute Data (AD).

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F19 Transfer Job Create (TJ)	M,H->E,reply
<i>Description</i>	
<p>The host requests that the equipment undertake one or more discrete (or atomic) transfers to achieve a host defined objective. The host provides the transfer specifications for each atomic transfer. Atomic transfers on separate ports on the equipment are allowed to execute in parallel. Atomic transfers for a port must be executed sequentially or in some cases concurrently. Both equipment transfer partners for each atomic transfer must receive appropriate Transfer Job Request messages in order to execute a transfer. If S4,F20 is multi-block, it must be preceded by the S4,F25/S4,F26 Inquire/Grant transaction.</p>	
<i>Structure</i>	
<pre> L, 2 1. &lt;DATAID&gt; 2. L, 2     1. &lt;TRJOBNAME&gt;     2. L, n                                [n = #atomic xfrs defined for this job]         1. L, 12                          [Specification for first atomic xfr]             1. &lt;TRLINK&gt;                    [Atomic transfer identifier]             2. &lt;TRPORT&gt;                    [Port to be used for transfer]             3. &lt;TROBJNAME&gt;                 [Transfer object identifier]             4. &lt;TROBJTYPE&gt;                 [Object type-what form is the material in]             5. &lt;TRROLE&gt;                    [Role in transfer-primary/secondary]             6. &lt;TRRCP&gt;                     [Transfer recipe identifier]             7. &lt;TRPTNR&gt;                    [Identifier of transfer partner]             8. &lt;TRPTPORT&gt;                  [Partner's Port to be Used]             9. &lt;TRDIR&gt;                     [Transfer direction-send or receive]             10. &lt;TRTYPE&gt;                   [Active or Passive]             11. &lt;TRLOCATION&gt;                 [Location to send/receive mtl]             12. &lt;TRAUTOSTART&gt;              [Does eqp await host start command after setup?]         .         .     n. L, 12                              [Specification for nth atomic xfr]         1. &lt;TRLINK&gt;         ↓         12. &lt;TRAUTOSTART&gt; </pre>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F20 Transfer Job Acknowledge (TJA)	S,H<-E
<i>Description</i>	
The equipment informs the host of its acceptance or rejection of the Transfer Job Request.	
<i>Structure</i>	
<pre> L, 3 1. &lt;TRJOBID&gt; 2. L,m          [m = number of atomic transfers in the transfer job.]    1. &lt;TRATOMICID<sub>1</sub>&gt;    .    .    m. &lt;TRATOMICID<sub>m</sub>&gt; 3. L,2    1. &lt;TRACK&gt;          [Accepted or rejected]    2. L,n          [n = # errors reported]       1. L,2          1. &lt;ERRCODE<sub>1</sub>&gt;          2. &lt;ERRTEXT<sub>1</sub>&gt;       .       .       n. L,2          1. &lt;ERRCODE<sub>n</sub>&gt;          2. &lt;ERRTEXT<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
A zero-length list (m = 0) is sent if the transfer job is rejected. A zero-length list (n = 0) is sent if the transfer job is accepted.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F21 Transfer Job Command (TC)	S,H->E,reply
<i>Description</i>	
This message is used by the host to modify a current transfer job on an equipment.	
<i>Structure</i>	
<pre> L, 3 1. &lt;TRJOBID&gt; 2. &lt;TRCMDNAME&gt;      [identifier of the transfer command] 3. L,n          [n = number of parameters = 0 if none]    1. L,2       1. &lt;CPNAME<sub>1</sub>&gt;      [transfer parameter name]       2. &lt;CPVAL<sub>1</sub>&gt;      [transfer parameter value]    .    .    n. L,2       1. &lt;CPNAME<sub>n</sub>&gt;       2. &lt;CPVAL<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F22 Transfer Command Acknowledge (TCA)	S,H<-E
<i>Description</i>	
Equipment accepts or rejects the transfer command.	
<i>Structure</i>	
L, 2 1. <TRACK> [Accepted or rejected] 2. L,n [n = # errors reported] 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If the command is accepted, n = 0.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F23 Transfer Job Alert (TJA)	S,H<-E,[reply]
<i>Description</i>	
Equipment informs the host that a transfer job milestone has been reached (e.g., job started or job complete). If complete, all equipment resources originally reserved for the transfer have been released.	
<i>Structure</i>	
L, 4 1. <TRJOBID> 2. <TRJOBNAME> 3. <TRJOBMS> 4. L, 2 1. <TRACK> [success or failure] 2. L,n [n = # errors reported] 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If the transfer job is completed successfully, n = 0.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F24 Transfer Alert Acknowledge (TLA)	S,H->E
<i>Description</i>	
Acknowledge receipt of the S4,F23 message.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F25 Multi-block Inquire (MB14)	S,H->E,reply
<i>Description</i>	
If a Stream 4 host-initiated message is more than a single block in length, this transaction must precede the message.	
<i>Structure</i>	
L, 2 1. <DATAID> 2. <DATALENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F26 Multi-block Grant (MBG4)	S,H<-E
<i>Description</i>	
Grant (or deny) permission to send multi-block message.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	

10.8.2.2 *Micro Level Messages* — The following messages support the equipment-to-equipment micro level handoff of material as defined in SEMI E32.

10.8.2.2.1 The messages which support the micro level are passed directly between the equipment. For the purpose of the communication link, one of the equipment must be designated the host and the other the equipment. The choice is up to the implementer. Equipment which are configurable to act as either host or equipment are suggested for ease of installation.

10.8.2.2.2 The two equipment involved in a micro level transfer assume different roles. One equipment is designated the “Primary Transfer Partner”, and the other is the “Secondary Transfer Partner” (see SEMI E32 for more definition). While some consistency of roles is expected, this designation is fluid and may change from one transfer to the next. The Primary Transfer Partner has more responsibility and thus initiates messages which the Secondary does not.

10.8.2.2.3 The selection of “Host” and “Equipment” for the communication link is not related to the fluid relationship of Primary and Secondary Transfer Partner. However, it is the designation of Primary or Secondary which determines the originator of certain messages. It is for this reason that the designation P = Primary and S = Secondary Transfer Partner.

#### 10.8.2.2.4 Micro Level Messages

Stream,Function Name (Mnemonic)	Direction
S4,F27 Handoff Ready (HR)	S,P<->S
<i>Description</i>	
Each transfer partner informs the other when they are ready to perform a specified atomic transfer. The TRLINK values from the two partners must match. The values contained in the atomic transfer specification pertain to the sender of the message (except where specified).	
<i>Structure</i>	
L, 2 1. <EQNAME> 2. L,11 [Specification for atomic xfr] 1. <TRLINK> [Atomic transfer identifier] 2. <TRPORT> [Port to be used for transfer] 3. <TROBJNAME> [Transfer object identifier] 4. <TROBJTYPE> [Object type-what form the material is in] 5. <TRROLE> [Role in transfer-primary/secondary] 6. <TRPTNR> [Identifier of transfer partner] 7. <TRPTPORT> [Partner's Port to be Used] 8. <TRDIR> [Transfer direction-send or receive] 9. <TRTYPE> [Active or Passive] 10. <TRLOCATION> [Location to send/receive mtl]	
<i>Exception</i>	
None	

#### S4,F28 Not Used

Stream,Function Name (Mnemonic)	Direction
S4,F29 Handoff Command (HC)	S,P->S
<i>Description</i>	
Command issued by the primary to the secondary transfer partner to achieve some physical action.	
<i>Structure</i>	
L, 4 1. <TRLINK> [Atomic Transfer identifier] 2. <MCINDEX> [Identified this specific Micro Cmd request] 3. <HOCMDNAME> [Requested Micro Cmd] 4. L,n [n = number of parameters] 1. L,2 1. <CPNAME <sub>1</sub> > [Micro Cmd parameter name] 2. <CPVAL <sub>1</sub> > [Micro Cmd parameter value] . . n. L,2 1. <CPNAME <sub>n</sub> > 2. <CPVAL <sub>n</sub> >	
<i>Exception</i>	
N = 0 if no parameters are used.	

#### S4,F30 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F31 Handoff Command Complete (HCC)	S,P<-S
<i>Description</i>	
Completion status of the micro command. This is sent from the secondary to the primary transfer partner when the command is completed or terminated.	
<i>Structure</i>	
L, 3 1. <TRLINK> [Atomic Transfer identifier] 2. <MCINDEX> [Links to specific micro command (S4,F31)] 3. L, 2 1. <HOACK> [success or failure] 2. L, n [n = # errors reported] 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
None	

#### S4,F32 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F33 Handoff Verified (HV)	P<->S
<i>Description</i>	
Sent by the primary transfer partner to inform the secondary that no more micro commands will be issued for this atomic transfer and to request a verification that the transfer is complete and successful. Also sent by the secondary partner following the receipt of this message to verify that the transfer is complete and successful (or to report problems).	
<i>Structure</i>	
L, 2 1. <TRLINK> 2. L, 2 1. <HOACK> [success or failure] 2. L, n [n = # errors reported] 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
None	

#### S4,F34 Not Used



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F35 Handoff Cancel Ready (HCR)	P<->S
<i>Description</i>	
Sent by either transfer partner to cancel a previous Handoff Ready message. This message is valid only before the handoff has begun.	
<i>Structure</i>	
<TRLINK>	
<i>Exception</i>	
None	

S4,F36 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F37 Handoff Cancel Ready Acknowledge (HCA)	P<->S
<i>Description</i>	
Sent by the receiver of the Handoff Cancel Ready message to accept or deny the cancel. The cancel request is denied if the handoff process has begun.	
<i>Structure</i>	
L, 2 1. <TRLINK> 2. <HOCANCELACK>	
<i>Exception</i>	
None	

S4,F38 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F39 Handoff Halt (HH)	P<->S
<i>Description</i>	
Sent by either transfer partner to cause all transfer related activity of the other to cease immediately. It is used when the equipment or material is at risk of damage.	
<i>Structure</i>	
<TRLINK>	
<i>Exception</i>	
None	

S4,F40 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S4,F41 Handoff Halt Acknowledge (HHA)	P<->S
<i>Description</i>	
Sent to equipment's transfer partner following completion of halt activities resulting from a previously received S4,F39.	
<i>Structure</i>	
L, 2 1. <TRLINK> 2. <HOHALTACK>	
<i>Exception</i>	
None	

S4,F42 Not Used

10.9 *Stream 5 Exception Handling* — This stream contains messages regarding binary and analog equipment exceptions. Exceptions are classified into two categories: errors and alarms. Messages S5,F1 through S5,F8 of this section provide basic alarm messages. The messages S5,F9 through S5,F18 provide extended capabilities for Exception Handling. When using messages F1–F8, alarms may be divided into categories as follows:

1. personal safety — Condition may be dangerous to people.
2. equipment safety — Condition may harm equipment.
3. parameter control warning — Parameter variation outside of preset limits—may harm product.
4. parameter control error — Parameter variation outside of reasonable control limits—may indicate an equipment malfunction.
5. irrecoverable error — Intervention required before normal use of equipment can resume.
6. equipment status warning — An unexpected condition has occurred, but operation can continue.
7. attention flags — A signal from a process program indicating that a particular step has been reached.
8. data integrity — A condition which may cause loss of data; usually related to Stream 6.

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

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F0 Abort Transaction (S5F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F1 Alarm Report Send (ARS)	S,H<-E,[reply]
<i>Description</i>	
This message reports a change in or presence of an alarm condition. One message will be issued when the alarm is set and one message will be issued when the alarm is cleared. Irrecoverable errors and attention flags may not have a corresponding clear message.	
<i>Structure</i>	
L, 3	
<ol style="list-style-type: none"> <li>1. &lt;ALCD&gt;</li> <li>2. &lt;ALID&gt;</li> <li>3. &lt;ALTX&gt;</li> </ol>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F2 Alarm Report Acknowledge (ARA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC5>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F3 Enable/Disable Alarm Send (EAS)	S,H->E,[reply]
<i>Description</i>	
This message will change the state of the enable bit in the equipment. The enable bit determines if the alarm will be sent to the host. Alarms which are not controllable in this way are unaffected by this message.	
<i>Structure</i>	
L, 2	
1. <ALED>	
2. <ALID>	
<i>Exception</i>	
A zero-length item for ALID means all alarms.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F4 Enable/Disable Alarm Acknowledge (EAA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC5>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F5 List Alarms Request (LAR)	S,H->E,reply
<i>Description</i>	
This message requests the equipment to send binary and analog alarm information to the host.	
<i>Structure</i>	
<ALID <sub>1</sub> , . . . ,ALID <sub>n</sub> >	
<i>Exception</i>	
A zero-length item means send all possible alarms regardless of the state of ALED.	





<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F6 List Alarm Data (LAD)	M,H<-E
<i>Description</i>	
This message contains the alarm data known to the equipment. There are “m” alarms in the list.	
<i>Structure</i>	
L, m 1. L, 3 1. <ALCD <sub>1</sub> > 2. <ALID <sub>1</sub> > 3. <ALTX <sub>1</sub> > 2. L, 3 . . m. L, 3 1. <ALCD <sub>m</sub> > 2. <ALID <sub>m</sub> > 3. <ALTX <sub>m</sub> >	
<i>Exception</i>	
If m = 0, no response can be made. A zero-length item returned for ALCD <sub>i</sub> or ALTX <sub>i</sub> means that value does not exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F7 List Enabled Alarm Request (LEAR)	S,H->E,reply
<i>Description</i>	
List alarms which are enabled.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F8 List Enabled Alarm Data (LEAD)	M,H<-E
<i>Description</i>	
This message is similar to S5,F6 except that it lists only alarms which are enabled.	
<i>Structure</i>	
Same as S5, F6	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F9 Exception Post Notify (EXPN)	S,H<-E,[reply]
<i>Description</i>	
This message provides the means to inform a host system that an exception condition is 'set'. Optionally, recovery actions for the exception may be sent.	
<i>Structure</i>	
L, 5 1. <TIMESTAMP> 2. <EXID> 3. <EXTYPE> 4. <EXMESSAGE> 5. L, n 1. <EXRECVRA <sub>1</sub> > . . . n. <EXRECVRA <sub>n</sub> >	
<i>Exception</i>	
A zero-length list (n = 0) shall be sent when there are no possible recovery actions. This is a single block message. The text in each of the EXRECVRA data items may need to be restricted in length to meet the single block requirement.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F10 Exception Post Confirm (EXPC)	S,H->E
<i>Description</i>	
Host confirms receipt of S5,F9 message from the equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F11 Exception Clear Notify (EXCN)	S,H<-E,[reply]
<i>Description</i>	
This message provides the means to inform a host system that an exception/alarm condition is no longer active (set).	
<i>Structure</i>	
L, 4 1. <TIMESTAMP> 2. <EXID> 3. <EXTYPE> 4. <EXMESSAGE>	
<i>Exception</i>	
EXMESSAGE can be used to provide the reason that the exception cleared.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F12 Exception Clear Confirm (EXCC)	S,H->E
<i>Description</i>	
Host confirms receipt of S5,F11 message from the equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F13 Exception Recover Request (EXRR)	S,H->E,reply
<i>Description</i>	
Request that the entity which is experiencing an error execute a recovery action.	
<i>Structure</i>	
L, 2 1. <EXID> 2. <EXRECVRA>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F14 Exception Recover Acknowledge (EXRA)	S,H<-E
<i>Description</i>	
The entity indicates a response to the recovery request.	
<i>Structure</i>	
L, 2 1. <EXID> 2. L, 2 1. <ACKA> 2. L, m                   (m = {0, 2}) 1. <ERRCODE> 2. <ERRTEXT>	
<i>Exception</i>	
The list m can be zero length, if the recovery request was accepted.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F15 Exception Recovery Complete Notify (EXRCN)	S,H<-E,[reply]
<i>Description</i>	
Allows the service provider to inform the controller/host that the recovery operation completed on a specific exception and an error code if the recovery terminated abnormally.	
<i>Structure</i>	
L, 3 1. <TIMESTAMP> 2. <EXID> 3. L, 2 1. <ACKA> 2. L, m                   (m = {0, 2}) 1. <ERRCODE> 2. <ERRTEXT>	
<i>Exception</i>	
This list m can be of zero length if the recovery was successful.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F16 Exception Recovery Complete Confirm (EXRCC)	S,H->E
<i>Description</i>	
Host confirms receipt of S5,F15 message from the equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F17 Exception Recovery Abort Request (EXRAR)	S,H->E,reply
<i>Description</i>	
Stop the recovery procedure on a specific exception.	
<i>Structure</i>	
1. <EXID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S5,F18 Exception Recovery Abort Acknowledge (EXRAA)	S,H<-E
<i>Description</i>	
Indicate the success of the request for Recovery Abort.	
<i>Structure</i>	
L, 2 1. <EXID> 2. L, 2 1. <ACKA> 2. L, m (m = {0, 2}) 1. <ERRCODE> 2. <ERRTEXT>	
<i>Exception</i>	
The list m can be of zero length if the abort was successful.	

10.10 *Stream 6 Data Collection* — This stream is intended to cover the needs of in-process measurements and equipment monitoring.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F0 Abort Transaction (S6F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F1 Trace Data Send (TDS)	M,H<-E,[reply]
<i>Description</i>	
<p>This function sends samples to the host according to the trace setup done by S2,F23. Trace is a time-driven form of equipment status.</p> <p>Even if S6,F1 is multi-block, it is not preceded by an Inquire/Grant transaction, because the Host S2,F23 is an implicit grant. Some equipment may support only single-block S6,F1, and may refuse an S2,F23 (Trace Initiate Send) message which would cause a multi-block S6,F1.</p>	
<i>Structure</i>	
<pre> L, 4 1. &lt;TRID&gt; 2. &lt;SMPLN&gt; 3. &lt;STIME&gt; 4. L, n    1. &lt;SV<sub>1</sub>&gt;    2. &lt;SV<sub>2</sub>&gt;    .    .    n. &lt;SV<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
A zero-length STIME means no value is given and that the time is to be derived from SMPLN along with knowledge of the request.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F2 Trace Data Acknowledge (TDA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC6>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F3 Discrete Variable Data Send (DVS)	M,H<-E,[reply]
<i>Description</i>	
Any data report which is initiated by an event, such as the completion of a measurement, rather than passage of time is called a discrete variable. Some equipment may have several possible events on which to send the data. S2,F15 is used to select the desired reporting events. Reports requiring only one block of data may report directly to the host with this message. If S6,F3 is multi-block, it must be preceded by the S6,F5/S6,F6 Inquire/Grant transaction.	
<i>Structure</i>	
<pre> L, 3   1. &lt;DATAID&gt;   2. &lt;CEID&gt;   3. L, n       1. L, 2           1. &lt;DSID<sub>1</sub>&gt;           2. L, m               1. L, 2                   1. &lt;DVNAME<sub>1</sub>&gt;                   2. &lt;DVVAL<sub>1</sub>&gt;               2. L, 2                   .                   .               m. L, 2                   1. &lt;DVNAME<sub>m</sub>&gt;                   2. &lt;DVVAL<sub>m</sub>&gt;           2. L, 2               .               .       n. L, 2           1. &lt;DSID<sub>n</sub>&gt;           2. etc. </pre>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F4 Discrete Variable Data Acknowledge (DVA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC6>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F5 Multi-block Data Send Inquire (MBI)	S,H<-E,reply
<i>Description</i>	
If the discrete data report S6F3, F9, F11, F13 can involve more than one block, this transaction must precede the transmission.	
<i>Structure</i>	
<pre> L, 2   1. &lt;DATAID&gt;   2. &lt;DATALENGTH&gt; </pre>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F6 Multi-block Grant (MBG)	S,H->E
<i>Description</i>	
Grant permission to send.	
<i>Structure</i>	
<GRANT6>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F7 Data Transfer Request (DDR)	S,H->E,reply
<i>Description</i>	
The host may initiate a data transfer of specified data stored in the equipment with this function.	
<i>Structure</i>	
<DATAID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F8 Data Transfer Data (DDD)	M,H<-E
<i>Description</i>	
Equipment sends data to the host.	
<i>Structure</i>	
Similar to the structure of S6,F3.	
<i>Exception</i>	
A zero-length item returned means the requested data cannot be sent.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F9 Formatted Variable Send (FVS)	M,H<-E,[reply]
<i>Description</i>	
The same function as S6,F3 except that the DVNAMEs are supplied from a predefined form that is known to the host. Thus, the data are more compact. If S6,F9 is multi-block, it must be preceded by the S6,F5/S6,F6 Inquire/Grant transaction.	
<i>Structure</i>	
L, 4 1. <PFCD> 2. <DATAID> 3. <CEID> 4. L,n 1. L,2 1. <DSID <sub>1</sub> > 2. L,m 1. <DVVAL <sub>1</sub> > . . m. <DVVAL <sub>m</sub> > 2. L,2 . . n. L,2 1. <DSID <sub>n</sub> > 2. etc.	

<i>Exception</i>
None

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F10 Formatted Variable Acknowledge (FVA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC6>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F11 Event Report Send (ERS)	M,H<-E, reply
<i>Description</i>	
<p>The purpose of this message is for the equipment to send a defined, linked, and enabled group of reports to the host upon the occurrence of an event (CEID).</p> <p>If S6,F11 is Multi-block, it must be preceded by the S6,F5/S6,F6 Inquire/Grant transaction.</p>	
<i>Structure</i>	
<pre> L, 3   1. &lt;DATAID&gt;   2. &lt;CEID&gt;   3. L, a       1. L, 2           1. &lt;RPTID<sub>1</sub>&gt;           2. L, b               1. &lt;V<sub>1</sub>&gt;               .               .               b. &lt;V<sub>b</sub>&gt;           .           .       a. L, 2          report a           1. &lt;RPTID<sub>a</sub>&gt;           2. L, c      #Vs this report               1. &lt;V<sub>1</sub>&gt;               .               .               c. &lt;V<sub>c</sub>&gt;         </pre>	
<i>Exception</i>	
<p>If there are no reports linked to the event a 'null' report is assumed. A zero-length list for # of reports means there are no reports linked to the given CEID.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F12 Event Report Acknowledge (ERA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC6>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F13 Annotated Event Report Send (AERS)	M,H<-E,reply
<i>Description</i>	
This message is the same as S6F11 with the exception that VID's are sent with data. If S6,F13 is Multi-block, it must be preceded by the S6,F5/S6,F6 Inquire/Grant transaction.	
<i>Structure</i>	
<pre> L, 3 1. &lt;DATAID&gt; 2. &lt;CEID&gt; 3. L, a     1. L, 2         1. &lt;RPTID<sub>1</sub>&gt;         2. L, b             1. L, 2                 1. &lt;VID<sub>1</sub>&gt;                 .                 .             b. L, 2                 1. &lt;VID<sub>b</sub>&gt;                 b. &lt;V<sub>b</sub>&gt;             .             .         a. L, 2             1. &lt;RPTID<sub>a</sub>&gt;             2. L, c                 1. L, 2                     1. &lt;VID<sub>1</sub>&gt;                     2. &lt;V<sub>1</sub>&gt;                 .                 .             c. L, 2                 1. &lt;VID<sub>c</sub>&gt;                 2. &lt;V<sub>c</sub>&gt; </pre>	
<i>Exception</i>	
If there are no reports linked to the event a 'null' report is assumed. A zero-length list for # of reports means there are no reports linked to the given CEID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F14 Annotated Event Report Acknowledge (AERA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC6>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F15 Event Report Request (ERR)	S,H->E, reply
<i>Description</i>	
The purpose of this message is for the host to demand a given report group from the equipment.	
<i>Structure</i>	
<CEID>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F16 Event Report Data (ERD)	M,H<-E
<i>Description</i>	
Equipment sends reports linked to given CEID to host.	
<i>Structure</i>	
Identical to structure of S6,F11.	
<i>Exception</i>	
A zero-length item means there are no reports linked to the given CEID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F17 Annotated Event Report Request (AERR)	S,H->E,reply
<i>Description</i>	
Same as S6,F15, but requests annotated reports.	
<i>Structure</i>	
<CEID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F18 Annotated Event Report Data (AERD)	M,H<-E
<i>Description</i>	
Equipment sends annotated reports linked to given CEID.	
<i>Structure</i>	
Same as S6,F13.	
<i>Exception</i>	
A zero-length item means there are no reports linked to the given CEID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F19 Individual Report Request (IRR)	S,H->E,reply
<i>Description</i>	
The purpose of this message is for the host to request a defined report from the equipment.	
<i>Structure</i>	
<RPTID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F20 Individual Report Data (IRD)	M,H<-E
<i>Description</i>	
Equipment sends variable data defined for the given RPTID to the host.	
<i>Structure</i>	
L,n	# of variable data items
1. <V <sub>1</sub> >	
.	
.	
n. <V <sub>n</sub> >	



Exception
None

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F21 Annotated Individual Report Request (AIRR)	S,H->E,reply
<i>Description</i>	
The purpose of this message is for the host to request an annotated defined report from the equipment.	
<i>Structure</i>	
<RPTID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F22 Annotated Individual Report Data (AIRD)	M,H<-E
<i>Description</i>	
Equipment sends annotated variable data defined for the given RPTID to the host.	
<i>Structure</i>	
L,n	# of variable data items
1. L,2	
1. <VID <sub>1</sub> >	
2. <V <sub>1</sub> >	
.	
.	
n. L,2	
1. <VID <sub>n</sub> >	
2. <V <sub>n</sub> >	
<i>Exception</i>	
A zero-length list for # of variable data items means RPTID is not defined.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F23 Request Spooled Data (RSD)	S,H->E,reply
<i>Description</i>	
The purpose of this message is for the host to request transmission or deletion of the messages currently spooled by the equipment.	
<i>Structure</i>	
<RSDC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F24 Request Spooled Data Acknowledgement Send (RSDAS)	S,H<-E
<i>Description</i>	
The purpose of this message is to acknowledge the receipt of the Request Spooled Data (S6,F23) and to respond with an appropriate acknowledge code.	
<i>Structure</i>	
<RSDA>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F25 Notification Report Send	M,H<->E,[reply]
<i>Description</i>	
<p>This message is used for change notifications or confirmation reports. A change notification is a report of an internal action and is not associated with a prior action requested by the host.</p> <p>A confirmation report is always associated with an earlier request for action. A confirmation report is sent to the initial requestor of a delayed action at the time the action is completed. A delayed action is an action that is any action not performed before the response to the initial request is sent. OPID contains the value of OPID in the initial request. LINKID is set to a non-zero value if and only if additional completion reports with the same OPID will be sent. If S6,F25 is multiblock, it must be preceded by the S6,F5/S6,F6 Inquire Grant transaction.</p>	
<i>Structure</i>	
<pre> L, 7 1. &lt;DATAID&gt; 2. &lt;OPID&gt; 3. &lt;LINKID&gt; 4. &lt;RCPSPEC&gt; 5. &lt;RMCHGSTAT&gt; 6. L, m     1. L, 2         1. &lt;RCPATTRID<sub>1</sub>&gt;         2. &lt;RCPATTRDATA<sub>1</sub>&gt;     .     .     m. L, 2         1. &lt;RCPATTRID<sub>m</sub>&gt;         2. &lt;RCPATTRDATA<sub>m</sub>&gt; 7. L, 2     1. &lt;RMACK&gt;     2. L, p         1. L, 2             1. &lt;ERRCODE<sub>1</sub>&gt;             2. &lt;ERRTEXT<sub>1</sub>&gt;         .         .         p. L, 2             1. &lt;ERRCODE<sub>p</sub>&gt;             2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
<p>OPID and LINKID are zero-length items when and only when S6,F25 is sent as a change notification rather than as a confirmation report. p = 0 if and only if RMACK indicates no errors.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F26 Notification Report Send Acknowledge	S,H<->E
<i>Description</i>	
<p>This message is used to acknowledge the confirmation report. It is defined for completeness and as an aid to the user in identifying problems.</p>	
<i>Structure</i>	
<ACKC6>	
<i>Exception</i>	
None	

Stream, Function Name (Mnemonic)	Direction
S6,F27 Trace Report Send (TRS)	M,H<-E,[reply]
<i>Description</i>	
The equipment sends a completed Trace Report to the host.	
<i>Structure</i>	
<pre> L, 3 1. &lt;DATAID&gt; 2. &lt;TRID&gt; 3. L,n          (n cannot exceed group size specified by S2F53)    1. L,p        (p is the number of reports for each trace sample)        1. L,2            1. &lt;RPTID<sub>1</sub>&gt;            2. L,m  (number of items in this data report)                1. &lt;V<sub>1</sub>&gt;                .                .                m. &lt;V<sub>m</sub>&gt;            .        p. L,2            1. &lt;RPTID<sub>p</sub>&gt;            2. L,m                1. &lt;V<sub>1</sub>&gt;                .                .                m. &lt;V<sub>m</sub>&gt;        .        .    n. L,p        (p is the number of reports for each trace sample)        1. L,2            1. &lt;RPTID<sub>1</sub>&gt;            2. L,m  (number of items in this data report)                1. &lt;V<sub>1</sub>&gt;                .                .                m. &lt;V<sub>m</sub>&gt;        .        .    p. L,2        1. &lt;RPTID<sub>p</sub>&gt;        2. L,m            1. &lt;V<sub>1</sub>&gt;            .            .            m. &lt;V<sub>m</sub>&gt; </pre>	
<i>Exception</i>	
The lists of variables associated with a unique RPTID are also unique. This structure illustrates the form of the message, so in general, V <sub>1</sub> for RPTID <sub>a</sub> and V <sub>1</sub> for RPTID <sub>b</sub> do not reference the same variable.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F28 Trace Report Send Acknowledge	S,H->E
<i>Description</i>	
The host Acknowledges receipt of the Trace Report.	
<i>Structure</i>	
<TRID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F29 Trace Report Request (TRR)	S,H->E
<i>Description</i>	
Request that the data reports assigned to the trace report be sampled and returned to the host.	
<i>Structure</i>	
<TRID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S6,F30 Trace Report Data (TRD)	M,H<-E
<i>Description</i>	
Message containing the requested data reports associated with the TRID of trace data report definition.	
<i>Structure</i>	
<pre> L, 3   1. &lt;TRID&gt;   2. L,n                                (n = number data reports defined for this TRID)     1. L,2       1. &lt;RPTID<sub>1</sub>&gt;       2. L,m                            (m = number of items in this RPTID)         1. &lt;V<sub>1</sub>&gt;         .         .         m. &lt;V<sub>m</sub>&gt;       .       .     n. L,2       1. &lt;RPTID<sub>n</sub>&gt;       2. L,m                            (m = number of items in this RPTID)         1. &lt;V<sub>1</sub>&gt;         .         .         m. &lt;V<sub>m</sub>&gt;       3. &lt;ERRCODE&gt;           </pre>	
<i>Exception</i>	
If TRID is unknown, a zero-length list (n = 0) shall be sent. Item 3 (ERRCODE) shall be set to zero length when there is no error.	

10.11 *Stream 7 Process Program Management* — The functions in this stream are used to manage and transfer process programs. Process programs are the equipment-specific descriptions that determine the procedure to be conducted on the material by a single piece of equipment. Methods are provided to transfer programs as well as establish the link between the process program and the material to be processed with that program.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F0 Abort Transaction (S7F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F1 Process Program Load Inquire (PPI)	S,H<->E,reply
<i>Description</i>	
This message is used to initiate the transfer of a process program or to select from stored programs. The message may be used to initiate the transfer of an unformatted process program (S7,F3/S7,F4) or a formatted process program (S7,F23/S7,F24), (S7,F31/S7,F32).	
<i>Structure</i>	
L, 2 1. <PPID> 2. <LENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F2 Process Program Load Grant (PPG)	S,H<->E
<i>Description</i>	
This message gives permission for the process program to be loaded.	
<i>Structure</i>	
<PPGNT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F3 Process Program Send (PPS)	M,H<->E,reply
<i>Description</i>	
The program is sent. If S7,F3 is multi-block, it must be preceded by the S7,F1/S7,F2 Inquire/Grant transaction.	
<i>Structure</i>	
L, 2 1. <PPID> 2. <PPBODY>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F4 Process Program Acknowledge (PPA)	S,H<->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F5 Process Program Request (PPR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the transfer of a process program.	
<i>Structure</i>	
<PPID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F6 Process Program Data (PPD)	M,H<->E
<i>Description</i>	
This message is used to transfer a process program.	
<i>Structure</i>	
L, 2 1. <PPID> 2. <PPBODY>	
<i>Exception</i>	
A zero-length list means request denied.	

NOTE 4: The equipment-to-host transfer of the process program, denoted by the R bit in the header (R = 1), provides the mechanism for the host computer to receive process programs created on the equipment. This allows use of the equipment without having process program generation capabilities on the host.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F7 Process Program ID Request (PIR)	S,H<-E,reply
<i>Description</i>	
This message is used to request the PPID for use on the material identified.	
<i>Structure</i>	
<MID>	
<i>Exception</i>	
None	





<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F8 Process Program ID Data (PID)	S,H->E
<i>Description</i>	
This message is used to transmit a single matrix entry in response to S7,F7.	
<i>Structure</i>	
L, 2 1. <PPID> 2. <MID>	
<i>Exception</i>	
A zero-length list returned means no such MID or other error.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F9 M/P M Request (MMR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the transmission of the material/process matrix. If the message is from the host, the response will be the current matrix in the equipment. If the message is from the equipment, the response will be a new matrix to initialize the equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

NOTE 5: M/PM defines the Material/Process Matrix. The Material/Process Matrix is a table which links the material to the process program to be used in processing the material.

Stream,Function Name (Mnemonic)	Direction
S7,F10 M/P M Data (MMD)	M,H<->E
<i>Description</i>	
In the response to S7,F9, the equipment will transmit the current matrix it contains. The matrix will be the sum of all matrix updates transmitted since initialization less the completed material whose linkages have been deleted. Programs with no pending material will be deleted from the matrix but not from the equipment program directory.	
<i>Structure</i>	
<pre> L,n                                number of process programs   1. L,2     1. &lt;PPID<sub>1</sub>&gt;     2. L,a                        (number of MID for this PPID)       1. &lt;MID<sub>1</sub>&gt;       .       .       a. &lt;MID<sub>a</sub>&gt;   2. L,2   .   .   n. L,2     1. &lt;PPID<sub>n</sub>&gt;     2. L,b       1. &lt;MID<sub>1</sub>&gt;       .       .       b. &lt;MID<sub>b</sub>&gt; </pre>	
<p>Function 10      Example 2 process programs (1 and 3 MID, respectively)</p> <pre> L,2   L,2     1. &lt;PPID<sub>1</sub>&gt;     2. L,1       1. &lt;MID<sub>a</sub>&gt;   L,2     1. &lt;PPID<sub>2</sub>&gt;     2. L,3       1. &lt;MID<sub>b</sub>&gt;       2. &lt;MID<sub>c</sub>&gt;       3. &lt;MID<sub>d</sub>&gt; </pre>	
<i>Exception</i>	
a = 0 indicates that this PPID will be used for all material processed. The last default transmitted will be the one used; all other entries will be deleted from the active matrix. A zero-length list returned means no such matrix.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F11 M/P M Update Send (UMS)	S,H->E,[reply]
<i>Description</i>	
This message is used by the host to add to the M/PM in the equipment.	
<i>Structure</i>	
L,n (number of process programs) 1. L,2 1. <PPID <sub>1</sub> > 2. L,a (number of MID's using PPID <sub>1</sub> ) 1. <MID <sub>1</sub> > . . a. <MID <sub>a</sub> > 2. L,2 . . n. L,2 1. <PPID <sub>n</sub> > 2. L,b 1. <MID <sub>1</sub> > . . b. <MID <sub>b</sub> >	
<i>Exception</i>	
If a = 0, then the preceding PPID is to be used for all material processed. All other entries will be deleted from the active matrix.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F12 M/P M Update Acknowledge (UMA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F13 Delete M/P M Entry Send (DES)	S,H->E,[reply]
<i>Description</i>	
This message is used to delete program to material linkages in the M/PM of the equipment.	
<i>Structure</i>	
L,n (number of process programs) 1. L,2 1. <PPID <sub>1</sub> > L,a (number of MID's using PPID) 1. <MID <sub>1</sub> > . . a. <MID <sub>a</sub> > 2. L,2 . . n. L,2 1. <PPID <sub>n</sub> > 2. L,b 1. <MID <sub>1</sub> > . . b. <MID <sub>b</sub> >	
<i>Exception</i>	
A delete consisting of a zero-length means delete all entries and generate a S7,F9 request to initialize matrix.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F14 Delete M/P M Entry Acknowledge (DEA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F15 Matrix Mode Select Send (MMS)	S,H->E,reply
<i>Description</i>	
This message is used by the host to change the method of process program selection in the equipment which might not support all modes.	
<i>Structure</i>	
<MMODE>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F16 Matrix Mode Select Acknowledge (MMA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

NOTE 6: The matrix structure allows the program linkages to be established for each MID or the multi-MID production plans for an extended period of time. The host system makes the choice of operating mode. By continuous updates to the equipment matrix, automatic system backup is achieved.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F17 Delete Process Program Send (DPS)	S,H->E,reply
<i>Description</i>	
This message is used by the host to request the equipment to delete process programs from equipment storage.	
<i>Structure</i>	
L, n (Number of process programs to be deleted) 1. <PPID <sub>1</sub> > . . n. <PPID <sub>n</sub> >	
<i>Exception</i>	
If n = 0, then delete all.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F18 Delete Process Program Acknowledge (DPA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F19 Current EPPD Request (RER)	S,H->E,reply
<i>Description</i>	
This message is used to request the transmission of the current equipment process program directory (EPPD). This is a list of all the PPIDs of the process programs stored in the equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F20 Current EPPD Data (RED)	M,H<-E
<i>Description</i>	
This message is used to transmit the current EPPD.	
<i>Structure</i>	
L, n (number of process programs in the directory) 1. <PPID <sub>1</sub> > . . n. <PPID <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F21 Equipment Process Capabilities Request (PCR)	S,H->E,reply
<i>Description</i>	
This message is used to request the Equipment Process Capabilities Data (PCD).	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

Stream,Function Name (Mnemonic)		Direction																																	
S7,F22 Equipment Process Capabilities Data (PCD)		M,H<-E																																	
<b>Description</b>																																			
This equipment message provides the information necessary for the host to create and partially verify the contents of a new process program or display the object of a process program previously generated by a host or equipment. The PCD defines the process program content acceptable to the originating equipment.																																			
<b>Structure</b>																																			
<pre> L, 5   1. &lt;MDLN&gt;   2. &lt;SOFTREV&gt;   3. &lt;CMDMAX&gt;   4. &lt;BYTMAX&gt;   5. L, c                                (c = Number of Possible Commands)       1. L, 11           1. &lt;CCODE&gt;           2. &lt;CNAME&gt;           3. &lt;RQCMD&gt;           4. &lt;BLKDEF&gt;           5. &lt;BCDS&gt;           6. &lt;IBCDS&gt;           7. &lt;NBCDS&gt;           8. &lt;ACDS&gt;           9. &lt;IACDS&gt;          10. &lt;NACDS&gt;          11. L, p                        (p = Number of Parameters)               1.                        (parameter specification) (see below)               .               .              p.                        (parameter specification)       2. L, 11       .       .       c. L, 11 </pre> <p>Parameter specifications depend on the data type of each parameter. The structure of each of the possible three groups is as follows:</p> <table> <thead> <tr> <th>Numeric Data</th><th>String Data</th><th>Boolean Data</th></tr> </thead> <tbody> <tr> <td>L, 9</td><td>L, 5</td><td>L, 4</td></tr> <tr> <td>1. &lt;PNAME&gt;</td><td>1. &lt;PNAME&gt;</td><td>1. &lt;PNAME&gt;</td></tr> <tr> <td>2. &lt;RQPAR&gt;</td><td>2. &lt;RQPAR&gt;</td><td>2. &lt;RQPAR&gt;</td></tr> <tr> <td>3. &lt;PDFLT&gt;</td><td>3. &lt;PDFLT&gt;</td><td>3. &lt;PDFLT&gt;</td></tr> <tr> <td>4. &lt;PMAX&gt;</td><td>4. &lt;PMAX&gt;</td><td>4. &lt;PMAX&gt;</td></tr> <tr> <td>5. &lt;LLIM&gt;</td><td>5. L, s</td><td></td></tr> <tr> <td>6. &lt;ULIM&gt;</td><td>1. &lt;STEMP<sub>1</sub>&gt;</td><td></td></tr> <tr> <td>7. &lt;UNITS&gt;</td><td>.</td><td></td></tr> <tr> <td>8. &lt;RESC&gt;</td><td>.</td><td></td></tr> <tr> <td>9. &lt;RESV&gt;</td><td>s. &lt;STEMP<sub>s</sub>&gt;</td><td></td></tr> </tbody> </table>			Numeric Data	String Data	Boolean Data	L, 9	L, 5	L, 4	1. <PNAME>	1. <PNAME>	1. <PNAME>	2. <RQPAR>	2. <RQPAR>	2. <RQPAR>	3. <PDFLT>	3. <PDFLT>	3. <PDFLT>	4. <PMAX>	4. <PMAX>	4. <PMAX>	5. <LLIM>	5. L, s		6. <ULIM>	1. <STEMP <sub>1</sub> >		7. <UNITS>	.		8. <RESC>	.		9. <RESV>	s. <STEMP <sub>s</sub> >	
Numeric Data	String Data	Boolean Data																																	
L, 9	L, 5	L, 4																																	
1. <PNAME>	1. <PNAME>	1. <PNAME>																																	
2. <RQPAR>	2. <RQPAR>	2. <RQPAR>																																	
3. <PDFLT>	3. <PDFLT>	3. <PDFLT>																																	
4. <PMAX>	4. <PMAX>	4. <PMAX>																																	
5. <LLIM>	5. L, s																																		
6. <ULIM>	1. <STEMP <sub>1</sub> >																																		
7. <UNITS>	.																																		
8. <RESC>	.																																		
9. <RESV>	s. <STEMP <sub>s</sub> >																																		
<b>Exception</b>																																			
None																																			

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F23 Formatted Process Program Send (FPS)	M,H<->E,reply
<i>Description</i>	
This message allows movement of formatted process programs between a piece of equipment and its host system. The values of MDLN and SOFTREV are obtained from the PCD used to generate the process program. If S7,F23 is multi-block, it must be preceded by the S7F1/F2 Inquire/Grant transaction.	
<i>Structure</i>	
<pre> L, 4   1. &lt;PPID&gt;   2. &lt;MDLN&gt;   3. &lt;SOFTREV&gt;   4. L, c                                (c = Number of Process Commands)       1. L, 2           1. &lt;CCODE&gt;           2. L, p                        (p = Number of Parameters)               1. &lt;PPARM<sub>1</sub>&gt;               .               .               p. &lt;PPARM<sub>p</sub>&gt;       2. L, 2       .       .       c. L, 2 </pre>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F24 Formatted Process Program Acknowledge (FPA)	S,H<->E
<i>Description</i>	
Acknowledges reception of a formatted process program at its destination and whether the process program was accepted by the interpreter. A returned status of "accepted" by the interpreter means only that the message is understood. The validity of the contents of the process program is determined through a separate transaction (S7,F27/S7,F28).	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F25 Formatted Process Program Request (FPR)	S,H<->E,reply
<i>Description</i>	
This message is used by either equipment or host to request a particular process program from the other.	
<i>Structure</i>	
<PPID>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F26 Formatted Process Program Data (FPD)	M,H<->E
<i>Description</i>	
This message transfers a process program in response to a request for the PPID. The values of MDLN and SOFTREV are obtained from the PCD used to generate the process program.	
<i>Structure</i>	
<pre> L, 4   1. &lt;PPID&gt;   2. &lt;MDLN&gt;   3. &lt;SOFTREV&gt;   4. L, c                                (c = Number of Process Commands)       1. L, 2           1. &lt;CCODE&gt;           2. L, p                        (p = Number of Parameters)               1. &lt;PPARM<sub>1</sub>&gt;               .               .               p. &lt;PPARM<sub>p</sub>&gt;       2. L, 2       .       .       c. L, 2 </pre>	
<i>Exception</i>	
A zero length list indicates the request was denied.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F27 Process Program Verification Send (PVS)	S,H<-E,reply
<i>Description</i>	
This message indicates to the host that a process program has been received and checked by the equipment. The result of the check is specified by the list of errors. An empty error list (list of zero-length) or a one-element list with ACKC7A having a value of zero (0) indicates no errors were found in the process program. The equipment may report as many errors as it seems appropriate. The equipment is responsible for sending a single copy of this message to the host after any reception of a formatted process program (S7,F23; S7,F26; S7,F31) or a large process program that was transferred via Stream 13 Data set Transfer Protocol (S7,F37; S7,F39; S7,F41; S7,F43). The verification of large unformatted process programs checks that the received process program is intact and was not corrupted by the Stream 13 transfer (e.g., by trying to load it). If S7,F27 is multi-block, it must be preceded by the S7,F29/S7,F30 Inquire/Grant Transaction.	
<i>Structure</i>	
<pre> L, 2   1. &lt;PPID&gt;   2. L, n                                (n = number of errors being reported)       1. L, 3           1. &lt;ACKC7A&gt;           2. &lt;SEQNUM&gt;           3. &lt;ERRW7&gt;       2. L, 3       .       .       n. L, 3 </pre>	
<i>Exception</i>	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F28 Process Program Verification Acknowledge (PVA)	S,H->E
<i>Description</i>	
Reply by host to equipment acknowledging reception of Process Program Verification Send (PVS).	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F29 Process Program Verification Inquire (PVI)	S,H<-E,reply
<i>Description</i>	
This message allows a piece of equipment to ask a host for permission to send a multi-block PVS.	
<i>Structure</i>	
<LENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F30 Process Program Verification Grant (PVG)	S,H->E
<i>Description</i>	
Reply by host to equipment providing response to Process Program Verification Inquire (PVI).	
<i>Structure</i>	
<PPGNT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F31 Verification Request Send (VRS)	M,H->E,reply
<i>Description</i>	
This message requests the interpreting equipment to check the contents of the provided process program and inform the host whether or not the process program is acceptable for processing at the machine. The values of MDLN and SOFTREV are obtained from the PCD used to generate the process program. If S7,F31 is multi-block, it must be preceded by the S7,F1/S7,F2 Inquire/Grant transaction.	
<i>Structure</i>	
L, 4 1. <PPID> 2. <MDLN> 3. <SOFTREV> 4. L, c (c = Number of Process Commands) 1. L, 2 1. <CCODE> 2. L, p (p = Number of Parameters) 1. <PPARM <sub>1</sub> > . . p. <PPARM <sub>p</sub> > 2. L, 2 . . c. L, 2	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F32 Verification Request Acknowledge (VRA)	S,H<-E
<i>Description</i>	
Acknowledges reception of a formatted process program verification request at its destination and whether the process program was accepted by the equipment. A returned status of accepted by the interpreter means only that the message is understood. The validity of the contents of the process program is specified through a separate transaction (S7,F27/S7,F28).	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F33 Process Program Available Request (PAR)	S,H<->E,reply
<i>Description</i>	
This message requests the interpreting host or equipment to check its process program library and tell the requester if the PPID will be supplied if requested.	
<i>Structure</i>	
<PPID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F34 Process Program Availability Data (PAD)	S,H<->E
<i>Description</i>	
This message allows originator to tell requester whether it can provide the specified process program and whether it can provide it formatted, unformatted, or both.	
<i>Structure</i>	
L, 3 1. <PPID> 2. <UNFLEN> 3. <FRMLEN>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F35 Process Program for MID Request (PPMR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the transfer of the process program to be used for the material identified.	
<i>Structure</i>	
<MID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F36 Process Program for MID Data (PPMD)	M,H<->E
<i>Description</i>	
This message is used to transfer the process program for the material identified.	
<i>Structure</i>	
L, 3 1. <MID> 2. <PPID> 3. <PPBODY>	
<i>Exception</i>	
A zero-length list returned means no such MID or other error.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F37 Large Process Program Send (LPPS)	S,H <-> E,reply
<i>Description</i>	
This is a request to send a process program via the Data Set Transfer protocol. The Data Set name, DSNAME, is the text string identifier of the process program, PPID. The Data Set is subsequently transferred as a Stream with the following internal SECSII structured data: <PPBODY>	
<i>Structure</i>	
<DSNAME>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F38 Large Process Program Acknowledge(LPPA)	S,H <-> E
<i>Description</i>	
Acknowledge or error. A returned status of “accepted” means only that the message is understood. When the receiving entity is the equipment, there is a separate verification transaction (S7,F27/S7,F28) that indicates the completion status of the request. When the receiving entity is the host, the completion of the request is signaled by an event report.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F39 Large Formatted Process Program Send (LFPPS)	S,H <-> E, reply
<i>Description</i>	
This is a request to send a formatted process program via the Data Set Transfer Protocol. The Data Set name, DSNAME, is the text string identifier of the process program, PPID. The Data Set is subsequently transferred as a Stream with the following internal SECSII structured data:	
L, 4 <ol style="list-style-type: none"> <li>1. &lt;PPID&gt;</li> <li>2. &lt;MDLN&gt;</li> <li>3. &lt;SOFTREV&gt;</li> <li>4. L, c <span style="float: right;">(c = Number of Process Commands)</span> <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;CCODE&gt;</li> <li>2. L, p <span style="float: right;">(p = Number of Parameters)</span> <ol style="list-style-type: none"> <li>1. &lt;PPARM<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> <li>p. &lt;PPARM<sub>p</sub>&gt;</li> </ol> </li> </ol> </li> <li>2. L, 2</li> <li>.</li> <li>.</li> <li>c. L, 2</li> </ol> </li> </ol>	
<i>Structure</i>	
<DSNAME>	
<i>Exception</i>	
none	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F40 Large Formatted Process Program Acknowledge (LFPPA)	S,H <-> E
<i>Description</i>	
Acknowledge or error. A returned status of “accepted” means only that the message is understood. When the receiving entity is the equipment, there is a separate verification transaction (S7,F27/S7,F28) that indicates the completion status of the request. When the receiving entity is the host, the completion of the request is signaled by an event report.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F41 Large Process Program Request(LPPR)	S,H <-> E, reply
<i>Description</i>	
This message is used to request the transfer of a process program via the Stream 13 Data set Transfer protocol. The Data Set name, DSNNAME, is the text string identifier of the process program, PPID. The Data Set is subsequently transferred as a Stream with the following internal SECSII structured data: <PPBODY>	
<i>Structure</i>	
<DSNAME>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F42 Large Process Program Acknowledge (LPPA)	S,H <-> E
<i>Description</i>	
Acknowledge or error. A returned status of “accepted” means only that the message is understood. When the receiving entity is the equipment, there is a separate verification transaction (S7,F27/S7,F28) that indicates the completion status of the request. When the receiving entity is the host, the completion of the request is signaled by an event report.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
It is possible to use the ACKC7 code “command will be performed with completion signaled later” for this message.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F43 Large Formatted Process Program Request (LFPPR)	S,H <-> E, reply
<i>Description</i>	
This message is used to request the transfer of a formatted process program via the Data set Transfer protocol. The Data Set name, DSNNAME, is the text string identifier of the process program, PPID. The Data Set is subsequently transferred as a Stream with the following internal SECSII structured data: L, 4 1. <PPID> 2. <MDLN> 3. <SOFTREV> 4. L,c (c = Number of Process Commands) 1. L,2 1. <CCODE> 2. L,p (p = Number of Parameters) 2. <PPARM <sub>1</sub> > . . p. <PPARM <sub>p</sub> > 2. L,2 . . c. L,2	
<i>Structure</i>	
<DSNAME>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S7,F44 Large Formatted Process Program Acknowledge (LFPPA)	S,H <-> E
<i>Description</i>	
Acknowledge or error. A returned status of “accepted” means only that the message is understood. When the receiving entity is the equipment, there is a separate verification transaction (S7,F27/S7,F28) that indicates the completion status of the request. When the receiving entity is the host, the completion of the request is signaled by an event report.	
<i>Structure</i>	
<ACKC7>	
<i>Exception</i>	
It is possible to use the ACKC7 code “command will be performed with completion signaled later” for this message.	

10.12 *Stream 8 Control Program Transfer* — The purpose of this stream is to provide the method for transmitting the programs used in the equipment to perform the control function or to execute the transmitted process program.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S8,F0 Abort Transaction (S8F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S8,F1 Boot Program Request (BPR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the transmission of the boot program. It is assumed that there is only one boot program associated with any given equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S8,F2 Boot Program Data (BPD)	M,H<->E
<i>Description</i>	
The boot program is required by some systems as a precursor to loading an operating system or executive program.	
<i>Structure</i>	
<BPD>	
<i>Exception</i>	
A zero-length item means no boot.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S8,F3 Executive Program Request (EPR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the executive program. It is assumed that there is only one executive program associated with any given equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S8,F4 Executive Program Data (EPD)	M,H<->E
<i>Description</i>	
The executive program is the master control program of the equipment. The executive may contain all the program required or it may contain the information required to request the rest of the program it needs on Stream 13.	
<i>Structure</i>	
<EPD>	
<i>Exception</i>	
None	

10.13 *Stream 9 System Errors* — This stream provides a method of informing the host that a message block has been received which cannot be handled or that a timeout on a transaction (receive) timer has occurred. The messages indicate either a Message Fault or a Communications Fault has occurred but do not indicate a Communications Failure has occurred.

10.13.1 *Communications Failure* — A Communications Failure occurs in a SECS-I environment when, and only when, the RTY limit is exceeded.

NOTE 7: In the event of a Communications Failure, no Stream 9 message is sent.

10.13.2 *Communications Fault* — A Communications Fault occurs when the equipment does not receive an expected message (when a transaction timer or a conversation timer has expired).

10.13.3 *Message Fault* — A Message Fault occurs when the equipment receives a message which it cannot process because of a fault that arises from the content, context, or length of the message.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F0 Abort Transaction (S9F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	





<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F1 Unrecognized Device ID (UDN)	S,H<-E
<i>Description</i>	
The device ID in the message block header did not correspond to any known device ID in the node detecting the error.	
<i>Structure</i>	
<MHEAD>	
<i>Exception</i>	
None	

#### S9,F2 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F3 Unrecognized Stream Type (USN)	S,H<-E
<i>Description</i>	
The equipment does not recognize the stream type in the message block header.	
<i>Structure</i>	
<MHEAD>	
<i>Exception</i>	
None	

#### S9,F4 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F5 Unrecognized Function Type (UFN)	S,H<-E
<i>Description</i>	
This message indicates that the function in the message ID is not recognized by the receiver.	
<i>Structure</i>	
<MHEAD>	
<i>Exception</i>	
None	

#### S9,F6 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F7 Illegal Data (IDN)	S,H<-E
<i>Description</i>	
<p>This message indicates that the stream and function were recognized, but the message format was incorrect. This message may be used to indicate that a message received by the equipment did not conform to § 10.3.1.3 “Compliance to Message Definitions” or that data within the received message was not an allowed data type.</p> <p>If the Message Definition allows for multiple formats for items or for multiple data types, this message may be used to indicate that the equipment does not support a format or type that was used in the instance of the message. Equipment suppliers are encouraged to document which data formats and data types are supported when multiple formats or types are allowed for items within a message.</p> <p>Similarly, if the message for a custom stream and function is not formatted according to the rules in § 10.3.1.3 for the Message Definition provided by the equipment supplier, this message may be used to indicate the message format was incorrect.</p>	
<i>Structure</i>	
<MHEAD>	
<i>Exception</i>	
None	

#### S9,F8 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F9 Transaction Timer Timeout (TTN)	S,H<-E
<i>Description</i>	
<p>This message indicates that a transaction (receive) timer has timed out and that the corresponding transaction has been aborted. It is up to the host to respond to this error in an appropriate manner to keep the system operational.</p>	
<i>Structure</i>	
<SHEAD>	
<i>Exception</i>	
None	

#### S9,F10 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F11 Data Too Long (DLN)	S,H<-E
<i>Description</i>	
<p>This message to the host indicates that the equipment has been sent more data than it can handle.</p>	
<i>Structure</i>	
<MHEAD>	
<i>Exception</i>	
None	

#### S9,F12 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S9,F13 Conversation Timeout (CTN)	S,H<-E
<i>Description</i>	
Data were expected but none were received within a reasonable length of time. Resources have been cleared.	
<i>Structure</i>	
L, 2 1. <MEXP> 2. <EDID>	
<i>Exception</i>	
None	

#### S9,F14 Not Used

10.14 *Stream 10 Terminal Services* — The functions of this stream is to pass textual messages between operator terminals attached to processing and/or testing equipment and the host. The equipment makes no attempt to interpret the text of the message, but merely passes it from terminal keyboard to the host or from the host to the display of the terminal. Management of human response times to information displayed on terminals is the responsibility of the host.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F0 Abort Transaction (S10F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F1 Terminal Request (TRN)	S,H<-E,[reply]
<i>Description</i>	
A terminal text message to the host.	
<i>Structure</i>	
L, 2 1. <TID> 2. <TEXT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F2 Terminal Request Acknowledge (TRA)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC10>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F3 Terminal Display, Single (VTN)	S,H->E, [reply]
<i>Description</i>	
Data to be displayed.	
<i>Structure</i>	
L, 2 1. <TID> 2. <TEXT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F4 Terminal Display, Single Acknowledge (VTA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC10>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F5 Terminal Display, Multi-Block (VTN)	M,H->E,[reply]
<i>Description</i>	
Data to be displayed on the equipment's terminal.	
<i>Structure</i>	
L, 2 1. <TID> 2. L,n 1. <TEXT <sub>1</sub> > . . n. <TEXT <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F6 Terminal Display, Multi-block Acknowledge (VMA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC10>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F7 Multi-block Not Allowed (MNN)	S,H<-E
<i>Description</i>	
An error message from a terminal that cannot handle a multi-block message from S10,F5.	
<i>Structure</i>	
<TID>	
<i>Exception</i>	
None	

#### S10,F8 Not Used

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F9 Broadcast (BCN)	S,H->E,[reply]
<i>Description</i>	
This function is generally the same as S10,F3 except that specific TID in each equipment need not be specified. Instead, the text is directed to each terminal in the equipment when the function is received. This function assumes that this feature exists on all equipment, otherwise repeated S10,F3 messages should be used.	
<i>Structure</i>	
<TEXT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S10,F10 Broadcast Acknowledge (BCA)	S,H<-E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<ACKC10>	
<i>Exception</i>	
None	

10.15 Stream 11 has been deleted and will not appear again in this publication.

10.15.1 It is the consensus of the Communications Committee that Stream 11 is obsolete. Its use is discouraged, and it has been removed from the 1989 edition of the standard. The reasons for removal are three-fold:

1. The purpose of this stream, as it was originally envisioned, is perceived to be of little use and can best be accomplished by other means beyond the scope of this standard;
2. The functions in this stream have many technical problems that severely limit their use;

3. There is a noticeable lack of implementations of this standard that utilize Stream 11 in its originally intended form.

NOTE 8: Applications that need to transfer unformatted data between the host and equipment should use the facilities of Stream 13.

10.16 *Stream 12 Wafer Mapping* — Messages which deal with coordinate positions and data associated with those positions. This includes functions such as wafer mapping with coordinates of die on a wafer and the associated binning information.

10.16.1 *Structure* — Functions 1–20 address the variations required by semiconductor equipment manufacturers in transmitting wafer maps to and from the process equipment (wafer probe through die attach). The functions include three basic formats. The three formats developed are:

1. Row/column format where a coordinate row starting position is given with die count in the row and starting direction. The respective binning information follows for each die.
2. Array format is structured such that a matrix array captures all or part of a wafer with the associated binning information.
3. Coordinate format provides an X/Y location and bin code for die on the wafer.

10.16.2 *Definitions and Descriptions* — The following information is required to perform map association to the physical wafer as it relates to the archival use and transmission of wafer maps.

1. Flat/Notch Location
2. Frame Rotation
3. Row Count
4. Column Count
5. Die Units of Measure
6. Die Size
7. Process Die Count
8. Reference Points
9. Bin Code Equivalents
10. Process Axis
11. Null Bin Code Value
12. ID Type

10.16.2.1 *Flat/Notch Location* — The position in degrees that the flat or notch are oriented during processing relative to a “normal” position of zero degrees. See Figure 6.

10.16.2.2 *Frame Rotation* — The orientation of a film frame relative to a “normal” position of zero degrees. See Figure 7.

10.16.2.3 *Row/Column Count* — The row and column counts are the total number of rows and columns, respectively, on a wafer which must be correlated directly with the wafer map. These numbers will always be greater than zero.

10.16.2.4 *Die Sizes* — The die size is given in standard units as specified by the die unit of measure item DUTMS, and will also be greater than zero. The value of the die size is determined by measuring the distance from a point on one die to the same point on the next die, often referred to as an index. This is depicted in the lower portion of Figure 7, Section B in the General Rules Section.

10.16.2.5 *Process Die Count* — The process die count item is used by equipment that is being map driven to make determinations about how much material to prepare. For example, a die attach will epoxy lead frames in advance of the attach process. By knowing the total number of die to be processed within a wafer map, the equipment can stop epoxying lead frames equivalent to the last die to be attached. This item is also used by the equipment to tell the host how many total die it processed for that map. For example, a die attach would use PRDCT to report the total die actually attached from a particular wafer.

10.16.2.6 *Reference Points* — Reference points provide a means of relating a map to the physical wafer. The total number of these points, and the method for assigning and detecting them, is the responsibility of the equipment. This standard only provides a means for transmitting them.

10.16.2.7 *Origin* — The origin is in one of five locations which is specified by the equipment when generating a wafer map. The origin is on an array structure having dimensional values equal to those specified by the row and column count. The origin then lies on one of the four corners of that array or in a center location determined by the following formula:

$$\left( \frac{\text{row} \sim \text{or} \sim \text{column} + 1}{2} \right) \text{truncated} \quad (1)$$

10.16.2.7.1 It is implicit in determining the center location that the upper-left-hand corner of the area, in the normal position, be counted as the first row and column position. An equipment requesting a map provides the origin location that it wants the map to be based on before transmission. If the equipment does not provide an origin, the host must provide a default value. An equipment transmitting a map must provide the origin with the map setup data.

10.16.2.8 *Bin Code Equivalents* — Bin code equivalents is a list of bin codes that the receiving equipment will process. (i.e., if a map contains codes 1 through 10 and the good die are bins 1 and 2, then bin code equivalent list could indicate 1 and 2 if only the good die categories were needed. These are the only bin codes to which an equivalent will drive for its respective process function.) In the case of X/Y coordinate format, the locations transmitted will be only those with the bin codes stated in the Bin Code Equivalent list, unless the length byte is set to zero, in which all of the bin codes in the map will be transmitted.

10.16.2.9 *Process Axis* — The process axis is the axis, either rows or columns, increasing or decreasing, and the side of the map, (top, bottom, left, or right, respectively) that the map data will originate from. This is based on the coordinate system as described under the General Rules section of this document.

10.16.2.10 *ID Type* — ID type indicates the appropriate material ID type (i.e., wafer, cassette, or film frame).

### 10.16.3 *General Rules*

10.16.3.1 *Map Data Size* — Stream 12 provides for the transmission of a complete map regardless of size. Equipment requiring segmented maps for transmission or reception will not be able to use the Stream 12 functions to handle the complete conversation.

10.16.3.2 *Orientation Conventions* — The orientation of a wafer presented for processing will differ from equipment to equipment. Stream 12 specifies conventions for expressing wafer orientation so that a map can be translated from one geometric representation to another.

10.16.3.2.1 The bottom of the wafer is the notch or the line of the major flat. The orientation of a wafer is measured in positive degrees clockwise (CW) from the “normal” position. The “normal” position is where the bottom of the wafer is closest to you when the wafer is lying horizontally in front of you with the die side facing up. The “normal” position has an orientation of zero degrees. See Figure 6 for graphic representation of wafer orientations.

10.16.3.2.2 The bottom of a film frame is also the notch or the line of notches. Its orientation and “normal” position are measured in the same manner as for wafers. See Figure 7 for examples of bottoms of film frames.

10.16.3.2.3 The orientation of an unmounted wafer presented for processing is given by the parameter FNLOC, Flat/Notch LOcation.

10.16.3.2.4 The ultimate orientation of a wafer presented for processing after it has been mounted on a film frame is the cumulative rotation of the wafer from the “normal” position on the film frame and the rotation of film frame as it is presented to the equipment. This is determined by the sum of the parameters FNLOC and FFROT, Film Frame ROTation. It is possible for an application to represent the ultimate orientation of a wafer in one of these parameters only and pass the other parameter as zero length.

10.16.3.2.5 Figure 6 shows wafers oriented at 270° with respect to the bottoms of a metal and round film frame. If one of these film frames were presented to an equipment rotated 90° clockwise (CW), (bottom facing the left edge of the page), the ultimate orientation of the wafer would be zero degrees.

10.16.3.2.6 In the case where either FNLOC or FFROT are unknown or irrelevant information, a zero-length data item is transmitted, and the item will be ignored by the application. One of the items must exist.

10.16.3.3 *Coordinate Axis System* — The coordinate axis orientation is shown in Figure 8, § A. The assumption is that the “X” or “column” coordinates increase to the right of the “Y-axis” and the “Y” or “row” coordinates increase above the “X-axis”. In describing the physical wafer it is also given that the coordinate axis orientation never rotates. The wafer moves or rotates within the coordinate axis system. The origin within the array describing the wafer’s coordinate system must be in one of five locations on that array (the center, upper-left, lower-left, upper-right, or lower-right corner of the array).

10.16.3.3.1 Figures 9 and 10 summarize the conversation protocol in the form of a flow chart. Since a single transmit inquire/grant can be used for one of three message function pairs, the application is required to examine MAPFT, is received as part of the map setup data to determine the appropriate function to follow. If the appropriate function is not transmitted, the conversation is aborted and the error is reported using the appropriate error reporting stream and function.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F0 Abort Transaction (S12,F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F1 Map Set-up Data Send (MSDS)	S,H<-E,reply
<i>Description</i>	
Used to send all of the map set-up data common to all formats and required to link the data map with the physical wafer.	
<i>Structure</i>	
L, 15 1. <MID> 2. <IDTYP> 3. <FNLOC> 4. <FFROT> 5. <ORLOC> 6. <RPSEL> 7. L, n 1. <REFP <sub>x</sub> REFP <sub>y</sub> > . . n. <REFP <sub>x</sub> REFP <sub>y</sub> > 8. <DUTMS> 9. <XDIES> 10. <YDIES> 11. <ROWCT> 12. <COLCT> 13. <NULBC> 14. <PRDCT> 15. <PRAXI>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F2 Map Set-up Data Acknowledge (MSDA)	S,H->E
<i>Description</i>	
Acknowledgment of receipt of complete set of map set-up parameters.	
<i>Structure</i>	
<SDACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F3 Map Set-up Data Request (MSDR)	S,H<-E,reply
<i>Description</i>	
Used to request set-up data from the host for the product ready to be processed at the equipment (common to all formats).	
<i>Structure</i>	
L, 9 1. <MID> 2. <IDTYP> 3. <MAPFT> 4. <FNLOC> 5. <FFROT> 6. <ORLOC> 7. <PRAXI> 8. <BCEQU> 9. <NULBC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F4 Map Set-up Data (MSD)	S,H->E
<i>Description</i>	
Used to send all of the map set-up data required to link the data map with the physical wafer.	
<i>Structure</i>	
L, 15 1. <MID> 2. <IDTYP> 3. <FNLOC> 4. <ORLOC> 5. <RPSEL> 6. L, n 1. <REFP <sub>x</sub> REFP <sub>y</sub> > . . n. <REFP <sub>x</sub> REFP <sub>y</sub> > 7. <DUTMS> 8. <XDIES> 9. <YDIES> 10. <ROWCT> 11. <COLCT> 12. <PRDCT> 13. <BCEQU> 14. <NULBC> 15. <MLCL>	
<i>Exception</i>	
A zero-length list returned means no such MID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F5 Map Transmit Inquire (MAPTI)	S,H<-E,reply
<i>Description</i>	
Used to prepare the host for map transmission. S12,F5 must precede all S12,F7–8, F9–10, & F11–12 transactions.	
<i>Structure</i>	
L, 4 1. <MID> 2. <IDTYP> 3. <MAPFT> 4. <MLCL>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F6 Map Transmit Grant (MAPTG)	S,H->E
<i>Description</i>	
Provides permission to transfer.	
<i>Structure</i>	
<GRNT1>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F7 Map Data Send Type 1 (MDS1)	M,H<-E,reply
<i>Description</i>	
Used to send map data from the equipment to the host in row or column compressed format. If S12,F7 is multi-block, it must be preceded by the S12,F5/S12,F6 Inquire/Grant transaction.	
<i>Structure</i>	
L, 3 1. <MID> 2. <IDTYP> 3. L,n 1. L, 2 1. <RSINF <sub>1</sub> > 2. <BINLT <sub>1</sub> > 2. L, 2 . . n. L, 2 1. <RSINF <sub>n</sub> > 2. <BINLT <sub>n</sub> >	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F8 Map Data Acknowledge Type 1 (MDA1)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<MDACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F9 Map Data Send Type 2 (MDS2)	M,H<-E,reply
<i>Description</i>	
Used to send map data from the equipment in array format. If S12,F9 is multi-block, it must be preceded by the S12,F5/S12,F6 Inquire/Grant transaction.	
<i>Structure</i>	
L, 4 1. <MID> 2. <IDTYP> 3. <STRP <sub>x</sub> STRP <sub>y</sub> > 4. <BINLT...>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F10 Map Data Acknowledge Type 2 (MDA2)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<MDACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F11 Map Data Send Type 3 (MDS3)	M,H<-E,reply
<i>Description</i>	
Used to send map data from the equipment in cartesian coordinate format. Bin values may or may not be included in the message. If S12,F11 is multi-block, it must be preceded by the S12,F5/S12,F6 Inquire/Grant transaction.	
<i>Structure</i>	
L, 3 1. <MID> 2. <IDTYP> 3. L, n 1. L, 2 1. <XYPOS1 <sub>x</sub> XYPOS1 <sub>y</sub> > 2. <BINLT1...> 2. L, 2 . . n. L, 2 1. <XYPOS <sub>n</sub> > 2. <BINLT <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F12 Map Data Acknowledge Type 3 (MDA3)	S,H->E
<i>Description</i>	
Acknowledge or error.	
<i>Structure</i>	
<MDACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F13 Map Data Request Type 1 (MDR1)	S,H<-E,reply
<i>Description</i>	
Used to request map data for product at equipment process station in row or column format.	
<i>Structure</i>	
L, 2 1. <MID> 2. <IDTYP>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F14 Map Data Type 1 (MD1)	M,H->E
<i>Description</i>	
Used to send map data from the host to the equipment in row or column format.	
<i>Structure</i>	
L, 3 1. <MID> 2. <IDTYP> 3. L, n 1. L, 2 1. <RSINF <sub>x1</sub> RSINF <sub>y1</sub> RSIN <sub>d</sub> > 2. <BINLT...> 2. L, 2 . . n. L, 2 1. <RSINF <sub>n</sub> > 2. <BINLT <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F15 Map Data Request Type 2 (MDR2)	S,H<-E,reply
<i>Description</i>	
Used to request map data for product at an equipment process station, in array format.	
<i>Structure</i>	
L, 2 1. <MID> 2. <IDTYP>	
<i>Exception</i>	
None	

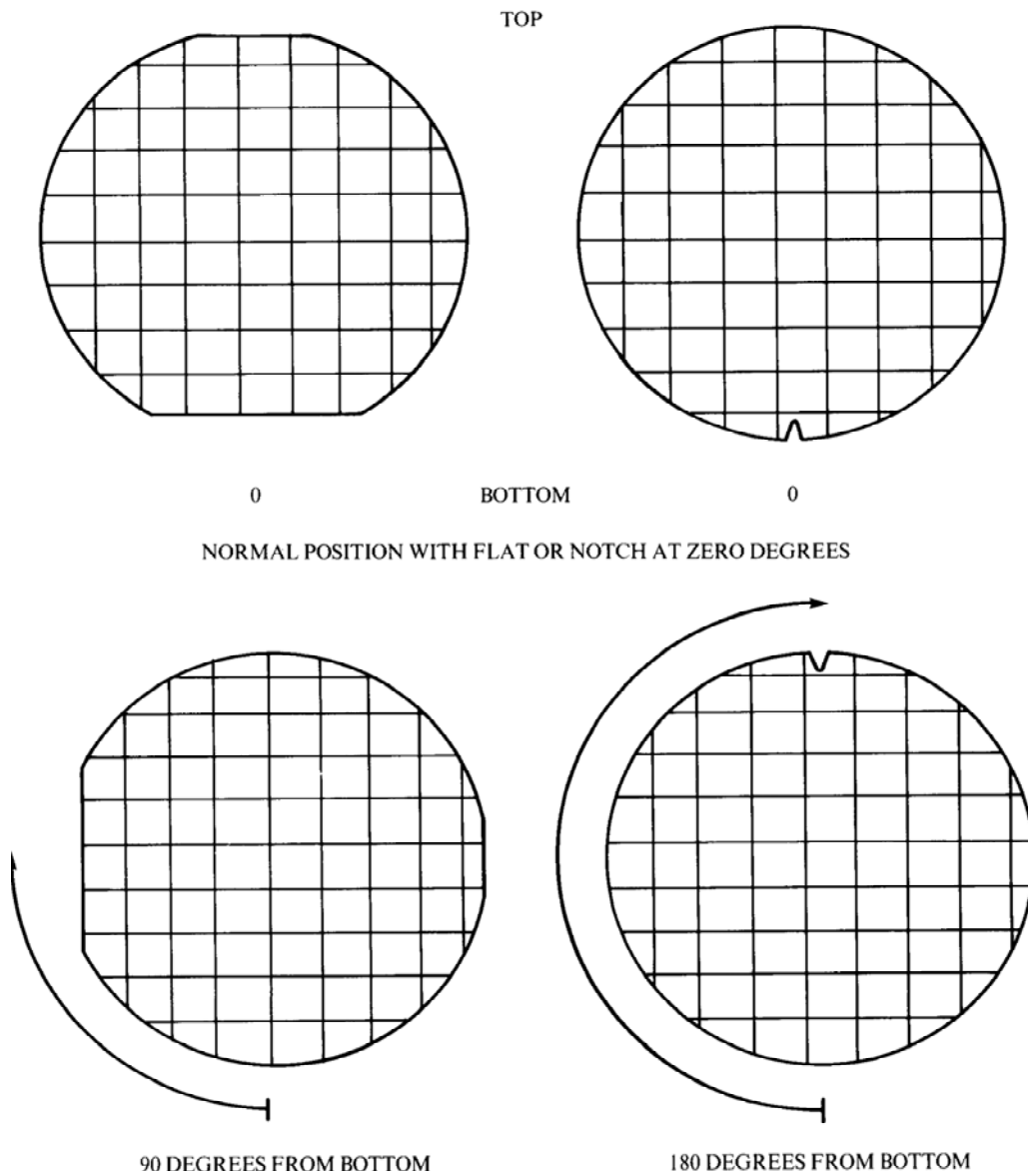
<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F16 Map Data Type 2 (MD2)	M,H->E
<i>Description</i>	
Used to send map data from the host to the equipment in array format.	
<i>Structure</i>	
L, 4 1. <MID> 2. <IDTYP> 3. <STRP <sub>x</sub> STRP <sub>y</sub> > 4. <BINLT>	
<i>Exception</i>	
A zero-length list returned means no such MID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F17 Map Data Request Type 3 (MDR3)	S,H<-E,reply
<i>Description</i>	
Used to request map data for product at an equipment process station in cartesian coordinate format.	
<i>Structure</i>	
L, 3 1. <MID> 2. <IDTYP> 3. <SDBIN>	
<i>Exception</i>	
None	

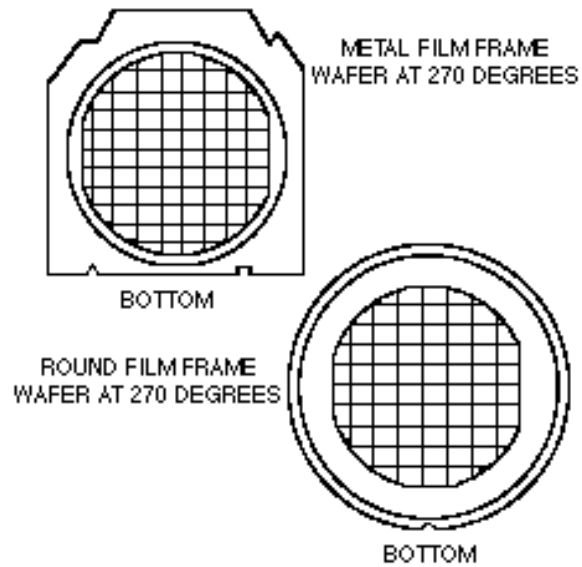
<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F18 Map Data Type 3 (MD3)	M,H->E
<i>Description</i>	
Used to send map data from the host to the equipment in cartesian coordinate format. Bin values may or may not be included.	
<i>Structure</i>	
L, 3 1. <MID> 2. <IDTYP> 3. L, n 1. L, 2 1. <XYPOS <sub>x1</sub> XYPOS <sub>y1</sub> > 2. <BINLT <sub>1</sub> . . . > 2. L, 2 . . n. L, 2 1. <XYPOS <sub>n</sub> > 2. <BINLT <sub>n</sub> >	
<i>Exception</i>	
A zero-length list returned means no such MID.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S12,F19 Map Error Report Send (MERS)	S,H<->E
<i>Description</i>	
Used to transmit map related errors.	
<i>Structure</i>	
L, 2 1. <MAPER> 2. <DATLC>	
<i>Exception</i>	
None	

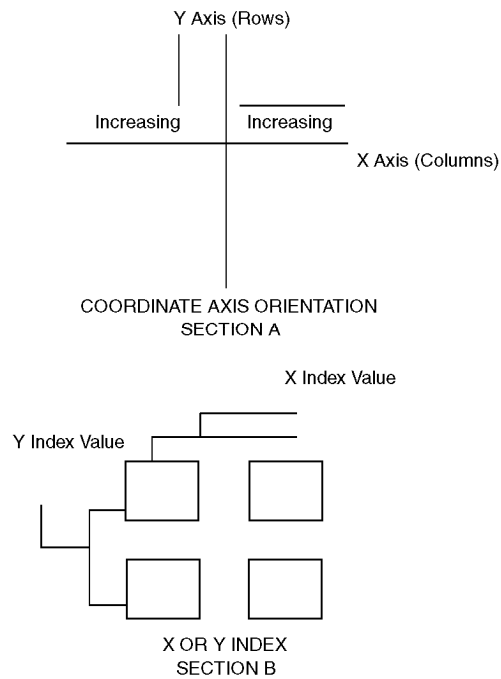
S12,F20 Not Used



**Figure 6**  
**Wafer Rotation Position in Degrees**

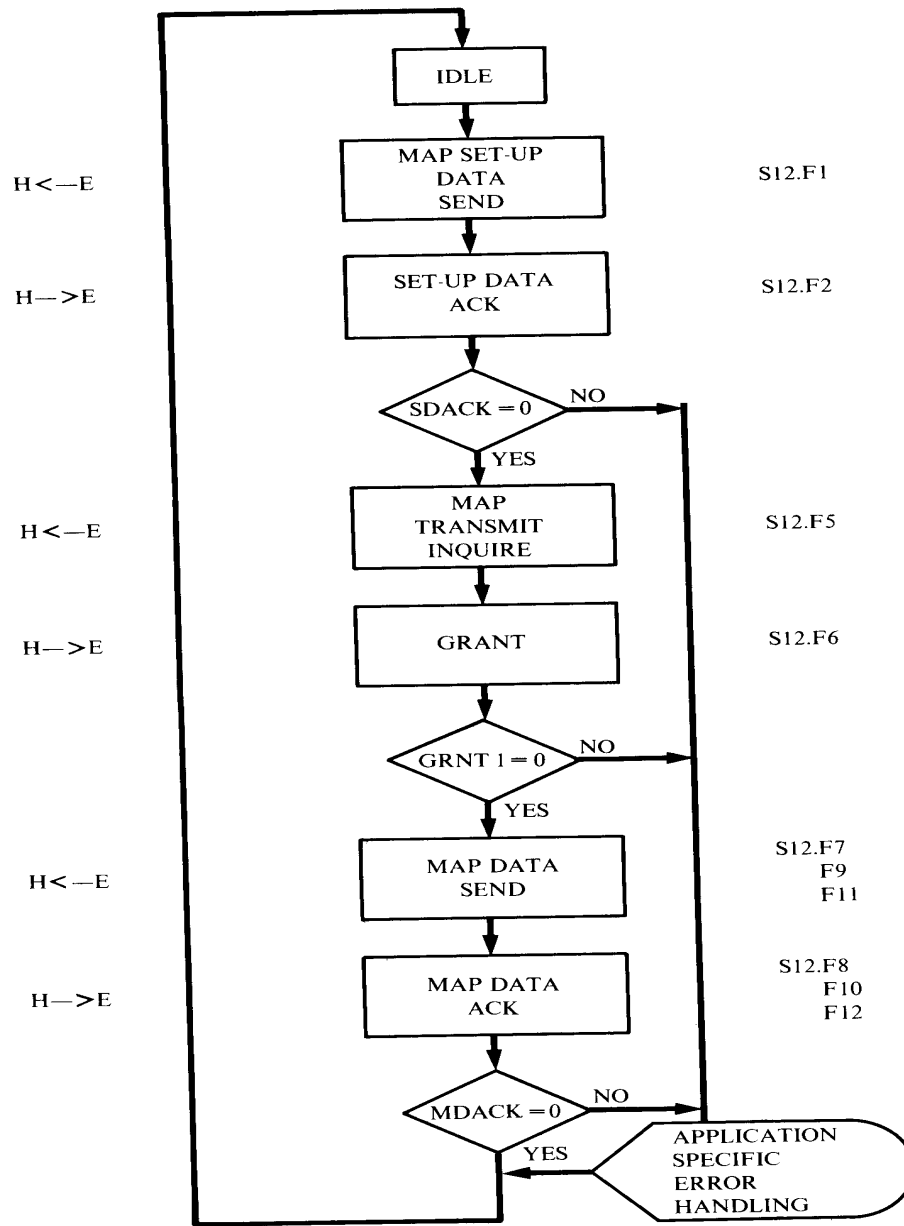


**Figure 7**  
**Wafer Rotation on Film Frame**

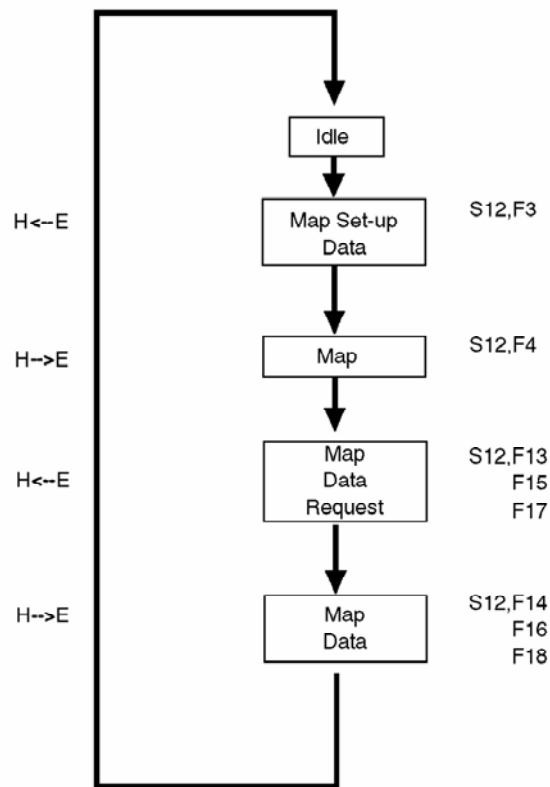


**Figure 8**  
**Orientation Reference and Index Determination**





**Figure 9**  
**Wafer Map Transmitted by Equipment**



**Figure 10**  
**Wafer Map Received by Equipment**

10.17 *Stream 13 Data Set Transfers* — This stream provides protocols to transfer data sets between systems. It is not intended to provide a general file access mechanism.

10.17.1 *Data Set Characteristics* — The data set may reside on the host or the equipment.

10.17.1.1 The term *data set* is used in a very general sense. A data set may represent a file, a data structure in memory, a collection of sensor values, or high density wafer profile data. The protocols define only the way data is sent from one system to another and do not define how the data set is stored by either the host or equipment.

10.17.1.2 The *sending* system is defined to be the system that has the data set. The *receiving* system is defined to be the system to which the data set is being transferred. The host or the equipment may assume either role.

10.17.2 *Unformatted Data Set Protocol* — The protocol for transferring unformatted data sets has the following characteristics:

1. Information about the record structure of the data set may be available.
2. ASCII records are transferred without the record terminating “noise” characters used by some operating systems.
3. Data sets do not need to be transferred in a single message.
4. No arbitrary limits are imposed on the length of one message. The maximum amount of data sent in each message is determined by both the sending and receiving systems, so there can be no data overruns.
5. There is a method of restarting a transfer in the event of an interruption.

10.17.2.1 *Data Set Name* — An ASCII string (format 20) which performs a function similar to the Message ID (Stream and Function) of the SECS-I protocol. This is a logical name which has meaning to both the equipment and the host. Neither the equipment nor the host is required to use the *Data Set Name* for the same information in any

other context. For example, maintenance data may be stored in the equipment in the file “WIDGET.DAT” and in the host as records in a database, but the *Data Set Name* may be “S11,F2”.

10.17.2.2 *Records* — The Record Type determines the way the data set is divided into messages for transfer to the receiving system. There are two types of records: Discrete and Stream.

1. A data set with *Discrete* records has a traditional record structure, such as ASCII text. *RecordLength* is the length of the longest record. Zero-length records are allowed. Each record from the data set is sent as a single item in a message.
2. If *Record Type* is *Stream*, then the data set has no internal structure which can be communicated with this protocol.

These kinds of data set might be, for example, a dump of main memory, SECS-II structured data, or data which has implicit record boundaries. *RecordLength* has no meaning for this kind of data set. Items containing data from the data set have no relationship to the structure of the data set.

10.17.2.3 *Transactions* — The basic data transfer is performed by the OPEN, READ, and CLOSE transactions. There is no explicit write transaction. A write is performed indirectly using the SEND transaction. A RESET transaction is provided to allow a graceful recovery after a crash. This protocol describes the transactions over the communications channel only. No assumption is made about the implementation of the transactions. For example, the OPEN transaction on a data set which is stored on a disk file does not necessarily cause the sending system to open the file.

10.17.2.3.1 The OPEN, READ, and CLOSE transactions are initiated by the receiving system. The SEND transaction is initiated by the sending system. The RESET transaction is initiated by either system. The usual transaction timer operates between the primary and secondary messages of each transaction. The time between transactions, especially between READ transactions, is application-specific so no additional timer is defined.

10.17.2.3.2 Internally, the protocol uses a *Handle* to keep track of multiple open data sets, and a *Checkpoint* which aids in error recovery. A value called *ReadLength* is used to negotiate the amount of data sent at one time.

10.17.2.4 *Handle* — Between the sending and receiving systems, more than one data set may be open at a time, or one data set may be opened many times. The Handle is a short name used to keep track of the state of a particular data set and instance of OPEN between the sending and receiving systems (see Figure 11). This *Handle* may be thought of as a name for a single application level connection from the sending to the receiving system. Its value is assigned in the primary message of the OPEN transaction.

10.17.2.4.1 The value used for the *Handle* must not be used in another OPEN by the receiving system to the same sending system until it is used in a CLOSE to that sending system, or the RESET transaction is sent by either system. For example, assume a host system opens a data set on equipment 255 using *Handle 1*. The host may not issue another OPEN to equipment 255 using *Handle 1* until it closes 1 on 255. However, the host may use *Handle 1* to open a data set on another piece of equipment, and equipment 255 may use *Handle 1* to open a data set on the host.

10.17.2.4.2 The number of data sets which may be open at one time and the number of times one data set may be opened is not specified by this standard. Error codes are defined for situations where the limits are exceeded. It must be possible to have one outstanding transaction (i.e., a primary message for which there has not been a reply) for each open *Handle*. If the sending system receives a primary message for a *Handle* which already has an outstanding transaction, then the error code for Pending Transaction is returned in the secondary message (see Figure 11).

10.17.2.5 *Checkpoint* — The response to each READ transaction contains the data and a new *Checkpoint* value. The *Checkpoint* is defined by and has meaning only for the sending system. Its purpose is to allow a data transfer to be restarted from the point of the last complete message after some communication interruption. The exact nature of the *Checkpoint* is not specified. It could be the byte index in the data set, a record counter (for Discrete records), or some other system-dependent value.

NOTE 9: Checkpoint and the SECS-II transaction timer define a performance requirement for the sending system. The sending system must be able to get data from any checkpoint location within a data set between the receipt of the OPEN primary message and the time for reply to the first READ.

10.17.2.5.1 The value of *Checkpoint* must conform to several rules:

1. The Checkpoint value is exactly four bytes long.

2. The beginning of the data set has Checkpoint value with all bits reset.
3. A Checkpoint with all bits set is illegal.
4. A Checkpoint supplied by the sending system which does not have all bits set is usable in an OPEN transaction to restart a data transfer without any lost data or duplicated data.

10.17.2.5.2 The receiving system defines the initial *Checkpoint* in the primary message of the OPEN transaction. The sending system returns the next *Checkpoint* in the response to each READ.

10.17.2.6 *Read Length* — The *Read Length* must be supplied by the receiving system with each READ transaction. It specifies the maximum number of data bytes which that system is prepared to process at one time. The sending system may supply less if it has limited resources. The sending system may supply more if *ReadLength* is zero, or is smaller than *RecordLength*.

10.17.2.7 *Reading a Data Set* — The basic data transfer is initiated with the OPEN transaction and completed with the CLOSE transaction. Information is sent from the sending to the receiving systems by a series of READ transactions.

10.17.2.8 *OPEN Transaction* — The receiving system sends a primary message containing the *DataSet Name* of the desired data set, the *Handle* to be used, and the *Checkpoint* of the initial READ transaction. The response from the sending system is a secondary message with a return code and the *RecordType* and *RecordLength* of the data set. If the return code is one of the error codes, then no data set was opened and the values of *RecordType* and *RecordLength* are undefined. If the *RecordType* is *Stream*, then the value of *RecordLength* is undefined. Notice that the undefined items will still appear in the secondary message.

10.17.2.8.1 The return code in the secondary message is one of the following:

- OK.
- ERROR: Unknown Data Set ID.
- ERROR: Try later (i.e., the data set is in use).
- ERROR: Too many open data sets.
- ERROR: Data set open too many times.
- ERROR: Handle in use.
- ERROR: Pending Transaction.

10.17.2.9 *The READ Transaction* — The receiving system sends a primary message which contains the *Handle*, and the *ReadLength*. The sending system responds with a secondary message which has a return code, the next *Checkpoint*, and zero or more items with data. At least one data item must be supplied unless there is an error. The return codes are:

- OK.
- ERROR: End of Data.
- ERROR: No open Data Set (i.e., incorrect Handle).
- ERROR: Cannot continue (i.e., a disk read error on the sending system).
- ERROR: Pending Transaction.

10.17.2.9.1 Any READ transaction which follows a READ which returned an error, except “Pending Transaction”, will generate the same error. The value of *Checkpoint* must be illegal (i.e., all bits must be set) when the “End of Data”, “No open Data Set”, or “Pending Transaction” error is returned. The value of *Checkpoint* error must be a value from which recovery may be attempted without duplicating data when the “Cannot continue” error is returned. Recovery may be attempted by issuing a CLOSE, followed by an OPEN with the last value of *Checkpoint*, and then another READ.

10.17.2.9.2 Each secondary message for the READ transaction must contain a whole number of *Discrete* records. A record may be sent as an ASCII or a binary item. *Stream* data sets are broken into pieces by the READ transaction without regard to internal structure. Each piece would be sent as a single binary item. The number of items which contain data depends on the *RecordType*, *Record Length*, and *Read Length*. The algorithm is designed so that the maximum length of the secondary message is deterministic. It gives the receiving system the ability to control the



amount of resources (such as SECS-I buffers) which it must allocate. The sending system may send less data than the maximum if it has limited resources. The performance (i.e., the packing of records into a message of some maximum size) should be very good for the case where records are all nearly the maximum length. This is assumed to be the usual case. The efficiency in pathological cases (e.g., many short records) will not be good, but the algorithm is robust enough to accommodate this without exceeding the maximum message size.

NOTE 10: If the *RecordType* is *Stream*, then there is exactly one item with a binary format whose length is not more than *ReadLength*. If the *RecordType* is *Discrete*, then the maximum number of items, *MaxItems*, is calculated by the formula:

$$\text{MaxItems} = \max \left( 1, \text{int} \left( \frac{\text{ReadLength}}{\text{RecordLength}} \right) \right) \quad (2)$$

10.17.2.9.3 The size of the secondary message may be less because of limited resources in the sending system.

10.17.2.9.4 For data sets with *Discrete* records, the format of each item is either ASCII (format 20) or binary (format 10). There is no requirement that all records be in the same format, but mixed record formats are not encouraged. Items with ASCII format should have only data characters. Characters which the sending system uses for control information (e.g., newline for a record terminator) should not appear. If an application finds it necessary to include these characters, then format 10, or *Stream* should be used.

10.17.2.10 *CLOSE Transaction* — This transaction terminates a data transfer and frees the *Handle* for future use. The primary message is sent by the receiving system and contains only the *Handle*. The sending system responds with a secondary message which has a return code.

10.17.2.10.1 The return code is one of the following:

- OK.
- ERROR: No open Data Set (i.e., incorrect *Handle*).
- ERROR: Pending Transaction.

10.17.2.11 *Sending a Data Set* — Writing a data set is performed by requesting that the receiving system read it. The sending system initiates the SEND transaction to request that a data set be read. The receiving system is expected to perform the OPEN, READ, and CLOSE transactions to transfer the data set if it accepts the request. The time between the secondary message of the SEND and the primary message of the OPEN depends on the application.

10.17.2.12 *The SEND Transaction* — The primary message sent from the sending system to the receiving system contains the *Data Set Name*.

10.17.2.12.1 The secondary message contains the *Data Set Name* and a return code which is one of the following:

- OK.
- ERROR: Unknown Data Set Name.
- ERROR: Try later (i.e., the system is busy).

10.17.2.13 *Error Recovery* — The receiving system may crash while a data set is open but no READ transaction is pending. The sending system will not be able to tell that this has happened because there is no timeout value defined between READ transactions. When the receiving system is restarted it may have forgotten which data sets were open. States in the two systems are now inconsistent.

10.17.2.14 *RESET Transaction* — The RESET transaction offers a way to resynchronize the two systems. When one system issues the primary message of the RESET transaction, it is informing the other system that any data sets which may have been open are to be closed. This applies to all data sets open between both systems. It is not necessary to issue CLOSE transactions for each individual data set because the RESET transaction is a global close.

10.17.2.15 Any equipment which uses Stream 13 must issue the RESET transaction as part of its initialization or bootstrap procedure. A host system must issue a RESET to equipment which uses Stream 13 during the initialization for that equipment.

10.17.2.16 *SECS-II Protocol Definition* — Figure 11 shows the state diagram for the sending system while a data set is being transferred. Each circle shows a possible state of the sending system. The names of these states are for

reference only. They are not meant to suggest an implementation. The arrows show transitions due to SECS-II messages received or sent by the sending system.

10.17.2.16.1 In the initial state (Idle) handle X is not open. The states marked with an asterisk are those in which a transaction is outstanding. If the sending system receives any primary message from the receiving system with handle X during the time it is in these states, then the secondary message for that transaction will contain the “Pending Transaction” error code, but the original transaction for handle X will not be affected. Some states, especially the error states, may take zero time in some implementations. In these cases, the “Pending Transaction” error code would not be returned from those states.

10.17.3 *Formatted Data Sets* — Formatted data sets are data sets transferred in a standard format. Stream 13 provides a method for transferring data sets in a table format. A table has both attributes and content. The attributes of the table provide information about the data set as a whole, such as the date and time that it was last modified, its size, etc. The content of the table consists of column headers and rows. A row is an ordered list of table elements. A column refers to all table elements at a specific position within all rows, where each column is identified by a corresponding text string as a column header. The table elements in the 1st column position are used as an identifier for the row.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F0 Abort Transaction (S13F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F1 Send Data Set Send (DSSS)	S,H<->E,reply
<i>Description</i>	
Sent by the sending system to request that the other system read a dataset.	
<i>Structure</i>	
L, 1	
1. <DSNAME>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F2 Send Data Set Acknowledge (DSSA)	S,H<->E
<i>Description</i>	
Sent by the receiving system in response to Send Data Set Send.	
L, 2	
1. <DSNAME>	
2. <ACKC13>	
<i>Structure</i>	
<i>Exception</i>	
The possible ACKC13 codes for this message are:	
0 = O.K.	
1 = ERROR:Try later.	
2 = ERROR:Unknown Data Set Name.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F3 Open Data Set Request (DSOR)	S,H<->E,reply
<i>Description</i>	
Sent by the receiving system to open a data set for reading.	
<i>Structure</i>	
L, 3 1. <HANDLE> 2. <DSNAME> 3. <CKPNT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F4 Open Data Set Data (DSOD)	S,H<->E
<i>Description</i>	
Sent by the sending system in response to Open Data Set Request.	
<i>Structure</i>	
L, 5 1. <HANDLE> 2. <DSNAME> 3. <ACKC13> 4. <RTYPE> 5. <RECLLEN>	
<i>Exception</i>	
The possible ACKC13 codes for this message are: 0 = O.K. 1 = ERROR:Try later. 2 = ERROR:Unknown Data Set Name. 3 = ERROR:Illegal Checkpoint value. 4 = ERROR:Too many open Data Sets. 5 = ERROR:Data set open too many times. 9 = ERROR:Handle in Use. 10 = ERROR:Pending Transaction.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F5 Read Data Set Request (DSRR)	S,H<->E,reply
<i>Description</i>	
Sent by the receiving system to read data from an open data set.	
<i>Structure</i>	
L, 2 1. <HANDLE> 2. <READLN>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F6 Read Data Set Data (DSRD)	M,H<->E
<i>Description</i>	
Sent by the sending system in response to Read Data Set Request.	
<i>Structure</i>	
L, 4 1. <HANDLE> 2. <ACKC13> 3. <CKPNT> 4. L,n 1. <FILDAT> . . n. <FILDAT>	
<i>Exception</i>	
The possible item formats, number of items (n), and length of each FILDAT item (lth) are given by the following table. MaxItems is defined in § 10.17.10.2.	
RTYPE	0 (Stream)                      1 (Discrete)
Item Format	10 (binary)                      10 (binary) or 20 (ASCII)
Maximum n	1                                  MaxItems
Maximum n	1 (ACKC13 = 0)                  1 (ACKC13 = 0)
	0(any error)                      0(any error)
Maximum  th	READLN                          RECLEN
Maximum  th	0                                  0
The possible ACKC13 codes for this message are: 0 = O.K. 6 = ERROR: No open Data Set. 7 = ERROR: Cannot Continue. 8 = ERROR: End of Data. 10 = ERROR: Pending Transaction.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F7 Close Data Set Send (DSCS)	S,H<->E,reply
<i>Description</i>	
Sent by the receiving system to close an open data set.	
<i>Structure</i>	
L, 1 1. <HANDLE>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F8 Close Data Set Acknowledge (DSCA)	S,H<->E
<i>Description</i>	
Sent by the sending system in response to Close Data Set Send (DSCS).	
<i>Structure</i>	
L, 2 1. <HANDLE> 2. <ACKC13>	
The possible ACKC13 codes for this message are: 0 = O.K. 6 = ERROR:No open Data Set. 10 = ERROR:Pending Transaction.	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F9 Reset Data Set Send (DSRS)	S,H<->E,reply
<i>Description</i>	
Sent by either system to close all open data sets.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F10 Reset Data Set Acknowledge (DSRA)	S,H<->E
<i>Description</i>	
Sent in response to Reset Data Set Send.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F11 Data Set Object Multi-Block Inquire (DSOMGI)	S,H<->E,reply
<i>Description</i>	
This message requests permission to send a multi-block data set. If the receiving system does not grant permission in the reply, the multi-block data set may not be sent. OBJSPEC is used to identify the data set object type and identifier and may include a destination. DATALENGTH represents the total message length, not the length of the data set.	
<i>Structure</i>	
L, 3 1. <DATAID> 2. <OBJSPEC> 3. <DATALENGTH>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F12 Data Set Object Multi-Block Grant (DSOMBG)	S,H<->E
<i>Description</i>	
This message grants or denies permission to send a multi-block data set.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	



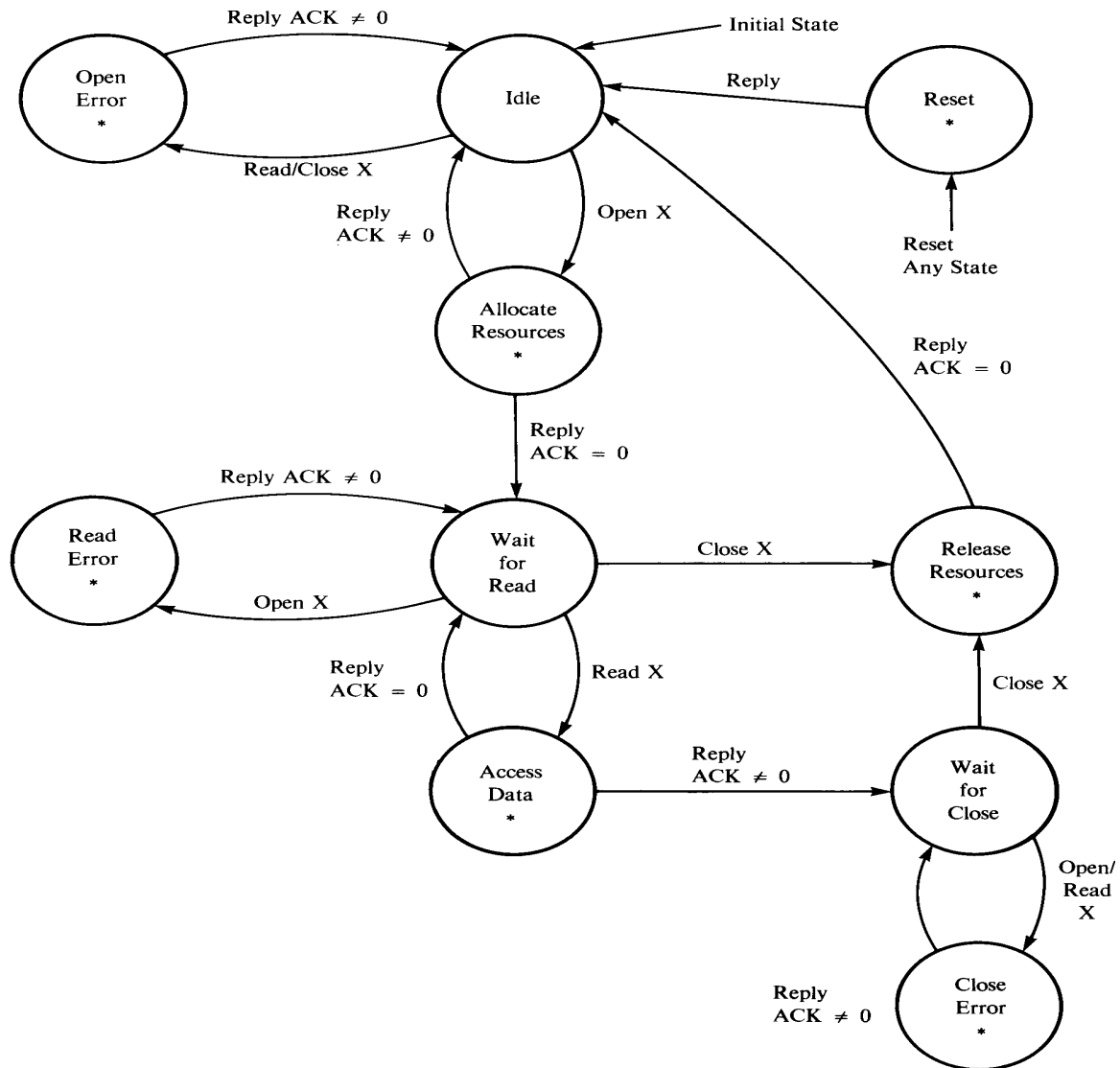
<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F13 Table Data Send (TDS)	M,H<->E,reply
<i>Description</i>	
This message allows the host and the equipment to exchange predefined datasets in a tabular format. The first element of every row is used to reference that row for all other elements. If S13,F13 is Multi-block, it must be preceded by the S13,F11/S13,F12 Inquire/Grant transaction.	
<i>Structure</i>	
L, 8 1. <DATAID> 2. <OBSPEC> 3. <TBLTYP> 4. <TBLID> 5. <TBLCMD> 6. L, n # of table attributes 1. L, 2 1. <ATTRID <sub>1</sub> > 2. <ATTRDATA <sub>1</sub> > . . n. L, 2 1. <ATTRID <sub>n</sub> > 2. <ATTRDATA <sub>n</sub> > 7. L, c # of column definitions 1. <COLHDR <sub>1</sub> > 1st column element description . . c. <COLHDR <sub>c</sub> > cth column element description 8. L, r # of row definitions 1. L, c <sub>1</sub> # of entries per definition 1. <TBLELT <sub>11</sub> > 1st table element, 1st row . m. <TBLELT <sub>1c<sub>1</sub></sub> > mth table element, 1st row . . r. L, c <sub>r</sub> rth row definition 1. <TBLELT <sub>r1</sub> > 1st table element, rth row . m. <TBLELT <sub>rc<sub>r</sub></sub> > mth table element, rth row	
<i>Exception</i>	
If OBSPEC is a zero-length item, then the owner of the table is the receiver of the message. If r is zero, any existing table definition of the given type and id is to be deleted. Otherwise, c <sub>1</sub> may not be zero, and the value of c <sub>1</sub> shall be less than or equal to the value of c.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F14 Table Data Acknowledge (TDA)	S,H<->E
<i>Description</i>	
This message is used to acknowledge the receipt of a table and to indicate any errors.	
<i>Structure</i>	
<pre> L, 2   1. &lt;TBLACK&gt;   2. L, p       1. L, 2           1. &lt;ERRCODE<sub>1</sub>&gt;           2. &lt;ERRTEXT<sub>1</sub>&gt;       .       .       p. L, 2           1. &lt;ERRCODE<sub>p</sub>&gt;           2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
p = 0 if and only if TBLACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S13,F15 Table Data Request (TDR)	M,H<->E,reply
<i>Description</i>	
This message allows the host or the equipment to request part or all of a specific table. Either specific columns or specific rows may be requested, but not both at the same time. If S13,F15 is Multi-block, it must be preceded by the S13,F11/S13,F12 Inquire/Grant transaction.	
<i>Structure</i>	
L, 7 1. <DATAID> 2. <OBJSPEC> 3. <TBLTYP> 4. <TBLID> 5. <TBLCMD> 6. L, p # of column definitions 1. <COLHDR <sub>1</sub> > 1st column element description . . p. <COLHDR <sub>p</sub> > pth column element description 7. L, q 1. <TBLELT <sub>1</sub> > 1st row identifier . . q. <TBLELT <sub>q</sub> >	
<i>Exception</i>	
If OBJSPEC is a zero-length item, then the owner of the table is the receiver of the message. Either p or q, or both, must be zero. If p = 0 and q = 0, all rows are requested; otherwise, only the specified columns, or the rows referenced by TBLELT, are requested.	

Stream,Function Name (Mnemonic)	Direction
S13,F16 Table Data (TD)	M,H<->E
<i>Description</i>	
This message is used to return data from the requested table.	
<i>Structure</i>	
<pre> L, 6   1. &lt;TBLTYP&gt;   2. &lt;TBLID&gt;   3. L, n                                     # of table attributes       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       n. L, 2           1. &lt;ATTRID<sub>n</sub>&gt;           2. &lt;ATTRDATA<sub>n</sub>&gt;   4. L, c                                     # of column definitions       1. &lt;COLHDR<sub>1</sub>&gt;                             1st column element description       .       .       c. &lt;COLHDR<sub>c</sub>&gt;                             cth column element description   5. L, r                                     # of row definitions       1. L, c<sub>1</sub>                                 # of entries per definition           1. &lt;TBLELT<sub>11</sub>&gt;                         1st table element, 1st row           .           .           m. &lt;TBLELT<sub>1c1</sub>&gt;                         last table element, 1st row           .           .           r. L, c<sub>r</sub>                             rth row definition               1. &lt;TBLELT<sub>r1</sub>&gt;                     1st table element, rth row               .               .               m. &lt;TBLELT<sub>rcr</sub>&gt;                     mth table element, rth row   6. L, 2       1. &lt;TBLACK&gt;       2. L, p           1. L, 2               1. &lt;ERRCODE<sub>11</sub>&gt;               2. &lt;ERRTEXT<sub>12</sub>&gt;           .           p. L, 2               1. &lt;ERRCODE<sub>p1</sub>&gt;               2. &lt;ERRTEXT<sub>p2</sub>&gt; </pre>	
<i>Exception</i>	
p = 0 if, and only if, TBLACK indicates no errors. The length c <sub>11</sub> of a table row may not exceed the value of c.	



**Figure 11**  
**The Sending System's State Diagram During a Data Set Transfer**  
 An Asterisk (\*) marks the states where a primary message which uses handle X would result in a secondary message with the error code for "Pending Transaction."

10.18 *Stream 14 Object Services* — The functions in this stream are used for generic functions concerning objects, including obtaining information about objects and setting values for an object.

<i>Stream, Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F0 Abort Transaction (S14F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F1 GetAttr Request (GAR)	S,H<->E,reply
<i>Description</i>	
<p>This message is used to request a set of specified attributes for one or more objects. It consists of an “object specifier” for the owner of the target objects (the objects of interest), the target object type, a list of identifiers of the target objects, a filter (a list of qualifying relationships) that limits the target objects of interest to those that meet all of the qualifications in and the specific attributes whose values are requested.</p> <p>The object specifier provides a specification of the owner of the target object(s). It contains a sequence of hierarchical object relationships. Each element of the object specifier identifies a specific object instance that is the superior of the following object instance in the sequence. The last object instance in the sequence is in a hierarchical relationship to the target objects. The target object type designates the type of the target object, and the list of object identifiers indicates the specific instance of that type that are of interest. The target type may be omitted only if object identifiers are unique across all object types and the list of identifiers is not empty.</p> <p>The object filter is an optional list of qualifications, each of which provides a condition to be applied to the object instances of interest. Each qualification objects of interest are those that meet all of the specified qualifications.</p> <p>The attribute relationship quantifier is a logical binary relationship ATTRRELN<sub>i</sub> that the specified qualifying value ATTRDATA<sub>i</sub> has to the corresponding attribute of each instance of the desired object type(s). The objects that are to be qualified with this filter have an attribute value V<sub>i</sub> such that the statement “ATTRDATA<sub>i</sub> ATTRRELN<sub>i</sub> V<sub>i</sub>” is TRUE. If ATTRRELN<sub>i</sub> is omitted, the relationship of equality is intended.</p> <p>For ASCII attribute values ATTRDATA<sub>i</sub>, the characters for question mark “?” and asterisk “*” are used as “wild characters” to provide filtering for certain object types. The character “?” may be used in any attribute or key attribute value with an ASCII format to represent “any single character” and may be repeated. The asterisk character “*” may be similarly used to represent a variable-length string, including a null string. The string “*x” represents a string of any length that ends in “x”, the string “x*” represents any string that begins with “x”, and the string “*” represents any string of any <u>non-zero</u> length. The comparison for text characters is case insensitive.</p> <p>Equipment is not required to support wild characters in particular, or attribute filters in general.</p>	
<i>Structure</i>	
<pre> L, 5   1. &lt;OBJSPEC&gt;   2. &lt;OBJTYPE&gt;   3. L, i                                i = identifiers of the object instances requested       1. &lt;OBJID<sub>1</sub>&gt;       .       .       i. &lt;OBJID<sub>i</sub>&gt;   4. L, q                                q = # object qualifiers to match       1. L, 3           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;           3. &lt;ATTRRELN<sub>1</sub>&gt;           .           .       q. L, 3           1. &lt;ATTRID<sub>q</sub>&gt;           2. &lt;ATTRDATA<sub>q</sub>&gt;           3. &lt;ATTRRELN<sub>q</sub>&gt;   5. L, a                                a = # attributes requested       1. &lt;ATTRID<sub>1</sub>&gt;       .       .       a. &lt;ATTRID<sub>a</sub>&gt; </pre>	
<i>Exception</i>	
<p>If OBJSPEC is a zero-length item, no object specifier is provided. If i = 0, only the filter is to be applied. If q = 0, no filter is specified. If both i and q = 0, information for all instances of the objects are requested. If a = 0, all attributes are requested.</p>	

Stream,Function Name (Mnemonic)	Direction
S14,F2 GetAttr Data (GAD)	M,H<->E
<i>Description</i>	
This message is used to transfer the set of requested attributes for the specified object(s). The order of attributes is retained from the primary message.	
<i>Structure</i>	
<pre> L, 2   1. L, n                                n = number of objects       1. L, 2           1. &lt;OBJID<sub>1</sub>&gt;           2. L, a                        a = number of attributes               1. L, 2                   1. &lt;ATTRID<sub>1</sub>&gt;                   2. &lt;ATTRDATA<sub>1</sub>&gt;               .               .           a. L, 2               1. &lt;ATTRID<sub>a</sub>&gt;               2. &lt;ATTRDATA<sub>a</sub>&gt;       .       .   n. L, 2       1. &lt;OBJID<sub>n</sub>&gt;       2. L, b                            b = number of attributes           1. L, 2               1. &lt;ATTRID<sub>1</sub>&gt;               2. &lt;ATTRDATA<sub>1</sub>&gt;           .           .       b. L, 2           1. &lt;ATTRID<sub>b</sub>&gt;           2. &lt;ATTRDATA<sub>b</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .       p. L, 2           1. &lt;ERRCODE<sub>p</sub>&gt;           2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no object specifier is provided. If n = 0, no objects matched the specified filter. If p = 0, no errors were detected.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F3 SetAttr Request (SAR)	S,H<->E, reply
<i>Description</i>	
This message is used to request that a given set of attributes be assigned specified values for all objects of the specified type and exactly matching the specified attribute requirements. Certain attributes may not be changed through the interface. For a description of filters, see S14,F1.	
<i>Structure</i>	
<pre> L, 4   1. &lt;OBJSPEC&gt;   2. &lt;OBJTYPE&gt;   3. L,i                               i = number of object instances requested       1. &lt;OBJID<sub>1</sub>&gt;       .       .       i. &lt;OBJID<sub>i</sub>&gt;   4. L,n                               n = # attribute settings       1. L,2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       n. L,2           1. &lt;ATTRID<sub>n</sub>&gt;           2. &lt;ATTRDATA<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no object specifier is provided.	

Stream,Function Name (Mnemonic)	Direction
S14,F4 SetAttr Data (SAD)	M,H<->E
<i>Description</i>	
This message is used to acknowledge that the attributes for the specified objects have been set as requested or to indicate an error for each attribute value that was not set as requested. The order of attributes is retained from the primary message.	
<i>Structure</i>	
<pre> L, 2   1. L, i                                i = number of objects requested     1. L, 2       1. &lt;OBJID<sub>1</sub>&gt;       2. L, n                            n = number of attributes set.         1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;         .         .         n. L, 2           1. &lt;ATTRID<sub>n</sub>&gt;           2. &lt;ATTRDATA<sub>n</sub>&gt;       .       .     i. L, 2       1. &lt;OBJID<sub>i</sub>&gt;       2. L, n         1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;         .         .         n. L, 2           1. &lt;ATTRID<sub>n</sub>&gt;           2. &lt;ATTRDATA<sub>n</sub>&gt;   2. L, 2     1. &lt;OBJACK&gt;     2. L, p                              p = number of errors reported       1. L, 2         1. &lt;ERRCODE<sub>1</sub>&gt;         2. &lt;ERRTEXT<sub>1</sub>&gt;       .       .       p. L, 2         1. &lt;ERRCODE<sub>p</sub>&gt;         2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0 for any object, the object was not found. If p = 0, no errors were detected.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F5 GetType Request (GTR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the types of objects owned by an object. This is an operation performed on an object type rather than on object instances. Wild characters “?” and “*” may be used as a filter for object types. Equipment is not required to support wild characters.	
<i>Structure</i>	
<OBJSPEC>	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no object specifier is provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F6 GetType Data (GTD)	
<i>Description</i>	
<i>Structure</i>	
<pre> L, 2   1. L, n                                n = number of object types       1. &lt;OBJTYP<sub>1</sub>&gt;       .       .       n. &lt;OBJTYP<sub>n</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, there are no owned object types. If p = 0, no errors were detected.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F7 GetAttrName Request (GANR)	S,H<->E,reply
<i>Description</i>	
This message is used to request the names of the attributes of specified types of owned objects. This is an operation performed on an object type rather than on object instances. Wild characters “?” and “*” may be used as a filter for object types. Equipment is not required to support wild characters.	
<i>Structure</i>	
<pre> L, 2   1. &lt;OBJSPEC&gt;   2. L, n                                n = # of object types       1. &lt;OBJTYP<sub>1</sub>&gt;       .       .       n. &lt;OBJTYP<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no object specifier is provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F8 GetAttrName Data (GAND)	M,H<->E
<i>Description</i>	
This message contains the names of the attributes of the requested objects.	
<i>Structure</i>	
<pre> L, 2   1. L, n                                n = number of object types       1. L, 2           1. &lt;OBJTYP<sub>1</sub>&gt;           2. L, a                        a = number of attributes               1. &lt;ATTRID<sub>1</sub>&gt;               .               .               a. &lt;ATTRID<sub>a</sub>&gt;           .           .       n. L, 2           1. &lt;OBJTYP<sub>n</sub>&gt;           2. L, b                        b = number of attributes               1. &lt;ATTRID<sub>1</sub>&gt;               .               .               b. &lt;ATTRID<sub>b</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no objects matched the specified filter. If p = 0, no errors were detected.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F9 Create Object Request (COR)	M,H<->E,reply
<i>Description</i>	
This message is used to request an object owner to create an object instance. OBJSPEC specifies the object owner.	
<i>Structure</i>	
<pre> L, 3   1. &lt;OBJSPEC&gt;   2. &lt;OBJTYPE&gt;   3. L,a                                a = # attributes requested       1. L,2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       a. L,2           1. &lt;ATTRID<sub>a</sub>&gt;           2. &lt;ATTRDATA<sub>a</sub>&gt; </pre>	
<i>Exception</i>	
If OBJSPEC is a null-length item, no object specifier is provided. If a = 0, no specific attribute settings are requested for the new object.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F10 Create Object Acknowledge (CAO)	M,H<->E
<i>Description</i>	
This message is used to acknowledge the success or failure of creating the new object specified. If successful, OBJSPEC is the object specifier of the new object. The list of attributes returned is dependent upon the type of object specified.	
<i>Structure</i>	
<pre> L, 3   1. &lt;OBJSPEC&gt;   2. L,b                                b = number of attributes returned       1. L,2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       b. L,2           1. &lt;ATTRID<sub>b</sub>&gt;           2. &lt;ATTRDATA<sub>b</sub>&gt;   3. L,2       1. &lt;OBJACK&gt;       2. L,p                            p = number of errors reported           1. L,2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L,2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	

<i>Exception</i>	
If OBJSPEC is a null-length item, no object was created. If b = 0, no attributes of the new object are returned. If p = 0, no errors were detected.	
<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F11 Delete Object Request	S,H<->E,reply
<i>Description</i>	
This message is used to request that the object specified in OBJSPEC be deleted. The list of attribute settings depends upon the type of object to be deleted.	
<i>Structure</i>	
<pre> L, 2   1. &lt;OBJSPEC&gt;   2. L, a                                n = # attribute settings       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       a. L, 2           1. &lt;ATTRID<sub>a</sub>&gt;           2. &lt;ATTRDATA<sub>a</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, no attribute settings are provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F12 Delete Object Acknowledge (DOA)	M,H<->E
<i>Description</i>	
This message is used to acknowledge the success or failure of deleting the object specified. The list of attributes returned is dependent upon the type of object to be deleted.	
<i>Structure</i>	
<pre> L, 2   1. L, b                                n = number of attributes returned       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       b. L, 2           1. &lt;ATTRID<sub>b</sub>&gt;           2. &lt;ATTRDATA<sub>b</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, no attribute values are returned. If p = 0, no errors were detected.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F13 Object Attach Request (OAR)	M,H<->E,reply
<i>Description</i>	
This message is sent by a supervisor to request the object specified in OBJSPEC to attach or reattach itself to the requestor.	
<i>Structure</i>	
<pre> L, 2   1. &lt;OBJSPEC&gt;   2. L, a                                a = # attribute settings       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       a. L, 2           1. &lt;ATTRID<sub>a</sub>&gt;           2. &lt;ATTRDATA<sub>a</sub>&gt; </pre>	
<i>Exception</i>	
If a = 0, no attribute settings are provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F14 Object Attach Acknowledge (OAA)	M,H<->E
<i>Description</i>	
This message is used to acknowledge the success or failure of the requested attachment. If successful, a non-zero token shall be returned for the supervisor's use in subsequent communications with the attached object.	
<i>Structure</i>	
<pre> L, 3   1. &lt;OBJTOKEN&gt;   2. L,b                      b = number of attributes       1. L,2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       b. L,2           1. &lt;ATTRID<sub>b</sub>&gt;           2. &lt;ATTRDATA<sub>b</sub>&gt;   3. L,2       1. &lt;OBJACK&gt;       2. L,p                  p = number of errors reported           1. L,2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L,2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
OBJTOKEN is zero if and only if p is non-zero. If b = 0, no attribute values are returned. If p = 0, no errors were detected.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F15 Attached Object Action Request (AOAR)	M,H<->E,reply
<i>Description</i>	
This message is used by a supervisor (only) to request an attached object to perform an action.	
<i>Structure</i>	
<pre> L, 4   1. &lt;OBJSPEC&gt;   2. &lt;OBJCMD&gt;   3. &lt;OBJTOKEN&gt;   4. L,a                      a = # attribute settings       1. L,2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       a. L,2           1. &lt;ATTRID<sub>a</sub>&gt;           2. &lt;ATTRDATA<sub>a</sub>&gt; </pre>	
<i>Exception</i>	
If a = 0, no attribute settings are provided.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F16 Attached Object Action Acknowledge (AOAA)	M,H<->E
<i>Description</i>	
This message is used to acknowledge the success or failure of an action requested by a supervisor.	
<i>Structure</i>	
<pre> L, 2   1. L, b                                b = number of attributes       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       b. L, 2           1. &lt;ATTRID<sub>b</sub>&gt;           2. &lt;ATTRDATA<sub>b</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If p = 0, no errors were detected.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F17 Supervised Object Action Request (SOAR)	S,H<->E,reply
<i>Description</i>	
This message is used to request a supervisor to have a supervised attached object perform an action. OBJSPEC specifies the supervisor, and TARGETSPEC specifies the attached object.	
<i>Structure</i>	
<pre> L, 4   1. &lt;OBJSPEC&gt;   2. &lt;OBJCMD&gt;   3. &lt;TARGETSPEC&gt;   4. L, a                                a = number of attribute settings       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       a. L, 2           1. &lt;ATTRID<sub>a</sub>&gt;           2. &lt;ATTRDATA<sub>a</sub>&gt; </pre>	
<i>Exception</i>	
If a = 0, no attribute settings are provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F18 Supervised Object Action Acknowledge (SOAA)	M,H<->E
<i>Description</i>	
This message is used to acknowledge the success or failure of an action requested of a supervisor.	
<i>Structure</i>	
<pre> L, 2   1. L, b                                b = number of attributes       1. L, 2           1. &lt;ATTRID<sub>1</sub>&gt;           2. &lt;ATTRDATA<sub>1</sub>&gt;       .       .       b. L, 2           1. &lt;ATTRID<sub>b</sub>&gt;           2. &lt;ATTRDATA<sub>b</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If b = 0, no attributes are returned. If p = 0, no errors were detected.	

Stream,Function Name (Mnemonic)	Direction
S14,F19 Generic Service Request (GSR)	M,H<->E, reply
<b>Description</b>	
The service requestor requests an object to perform the specified service with its associated parameters. If multi-block, it shall be preceded by the S14F23/F24 Multi-Block Inquire/Grant transaction. DATAID is given uniquely to each message. OPID is uniquely specified to identify delayed completion information for time consuming service. OPID could be zero if and only if the service cannot take a long time.	
<b>Structure</b>	
L, 5 <ol style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;OPID&gt;</li> <li>3. &lt;OBSPEC&gt;</li> <li>4. &lt;SVCNAME&gt;</li> <li>5. L, m # of parameter groups <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>1</sub>&gt; service parameter 1 name</li> <li>2. &lt;SPVAL<sub>1</sub>&gt; service parameter 1 value</li> </ol> </li> <li>2. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>2</sub>&gt; service parameter 2 name</li> <li>2. &lt;SPVAL<sub>2</sub>&gt; service parameter 2 value</li> </ol> </li> <li>.</li> <li>.</li> <li>m. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>m</sub>&gt; service parameter m name</li> <li>2. &lt;SPVAL<sub>m</sub>&gt; service parameter m value</li> </ol> </li> </ol> </li> </ol>	
If a specific value of SPNAME is defined to have a SPVAL defined as a LIST, it shall always be a LIST. If the SPVAL that is associated to that specific value of SPNAME is defined to be anything other than LIST, it will result in a format error.	
<b>Exception</b>	
A zero length list, m = 0, indicates that no parameter groups are sent with the service request. OBSPEC can be a null length item if no object provide the services is defined in the standards which are referred to and it is assumed that "Service Provider" is delegated and handled as if it is an object.	
Notes:	
1. If some service parameters are attributes of the specified object, service parameter name-value pair, that is SPNAME and SPVAL, is actually attribute id-data pair, that is ATTRID and ATTRDATA. An example of parameter part in the message format could be interpreted as below.	
L, m <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>1</sub>&gt;</li> <li>2. &lt;SPVAL<sub>1</sub>&gt;</li> </ol> </li> <li>2. L, 2</li> <li>.</li> <li>.</li> <li>k. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ATTRID<sub>k</sub>&gt;</li> <li>2. &lt;ATTRDATA<sub>k</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>m.</li> </ol>	
2. If SPVAL is a LIST, the items that make up that list shall take on one of the following forms: (1) a list of items with an identical format, (2) a LIST of SPNAME, SPVAL pairs, as illustrated below. When SPVAL is actually ATTRDATA, even if it is a LIST, it or its parts are not required to expand into lower level items if their names have not been formally named in the corresponding SEMI standard.	

A) L, 2 1. <SPNAME <sub>a</sub> > 2. L, m 1. <SPVAL <sub>a1</sub> > 2. <SPVAL <sub>a2</sub> > . . m. <SPVAL <sub>am</sub> >	B) L, 2 1. <SPNAME <sub>b</sub> > 2. L, n 1. L, 2 1. <SPNAME <sub>b1</sub> > 2. <SPVAL <sub>b1</sub> > . . n. L, 2 1. <SPNAME <sub>bn</sub> > 2. <SPVAL <sub>bn</sub> >
--	---

Stream, Function Name (Mnemonic)	Direction
S14,F20 Generic Service Acknowledge (GSA)	M,H<->E
<i>Description</i>	
<p>The service provider acknowledges requested service or reports any error(s). If the service is accepted and completed at once, (i.e., SVCACK = 0) then, the required parameters for the response are listed within this message. If the service is accepted but the performance of the service takes a relatively longer time due to some physical reason, (i.e., SVCACK = 4) then, the service response will be sent with the required parameters when the service is completed (see S14,F21/F22). The determination as to whether a delayed response is acceptable is dependent upon the implementation. LINKID is set to a non-zero value if and only if additional completion reports will be sent. The data items are mapped to the service response parameter list of the applied object service. Parameters that apply to the list are described in the standard that defines the application of the service.</p>	
<i>Structure</i>	
L, 4 1. <SVCACK> 2. <LINKID> 3. L, n # of parameter groups 1. L, 2 1. <SPNAME <sub>1</sub> > service parameter 1 name 2. <SPVAL <sub>1</sub> > service parameter 1 value 2. L, 2 1. <SPNAME <sub>2</sub> > service parameter 2 name 2. <SPVAL <sub>2</sub> > service parameter 2 value . . n. L, 2 1. <SPNAME <sub>n</sub> > service parameter n name 2. <SPVAL <sub>n</sub> > service parameter n value 4. L, 2 1. <SVCACK> 2. L, p p = number of errors reported 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2	

1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
<p>If p = 0, no errors were detected. If n = 0 parameters are returned. SVCACK is specified twice in the above structure for backward compatibility with the previous structure of this message, and for consistency with other stream 14 reply messages. The values of both SVCACK data items in this structure must be equal. The above message structure (L,4) is the preferred structure for this message and must be used for new implementations. The following message structure (L,3) is included for compatibility with previous implementations and must not be used for new implementations.</p> <p>Structure:</p> <p>L, 3</p> <ol style="list-style-type: none"> <li>1. &lt;SVCACK&gt;</li> <li>2. &lt;LINKID&gt;</li> <li>3. L, n <span style="float: right;"># of parameter groups</span> <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>1</sub>&gt; <span style="float: right;">service parameter 1 name</span></li> <li>2. &lt;SPVAL<sub>1</sub>&gt; <span style="float: right;">service parameter 1 value</span></li> </ol> </li> <li>2. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>2</sub>&gt; <span style="float: right;">service parameter 2 name</span></li> <li>2. &lt;SPVAL<sub>2</sub>&gt; <span style="float: right;">service parameter 2 value</span></li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 2 <ol style="list-style-type: none"> <li>1. &lt;SPNAME<sub>n</sub>&gt; <span style="float: right;">service parameter n name</span></li> <li>2. &lt;SPVAL<sub>n</sub>&gt; <span style="float: right;">service parameter n value</span></li> </ol> </li> </ol> </li> </ol>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F21 Generic Service Completion Information (GSCI)	M,H<->E, reply
<i>Description</i>	
<p>The service provider notifies the original service requestor when requested service on an object is completed, either successfully or unsuccessfully. The service provider may send required information using reply parameters. If the service was accepted and completed at once, the information was carried with the acknowledge message, that is secondary message of requesting one, and this message is not provided.</p> <p>OPID contains the value of OPID in the initial request (i.e., S14F19). LINKID is set to a non-zero value if and only if additional completion reports with the same OPID will be sent. If multi-block, it shall be preceded by the S14F23/F24 Multi-Block Inquire/Grant transaction.</p>	
<i>Structure</i>	
<pre> L, 5 1. &lt;DATAID&gt; 2. &lt;OPID&gt; 3. &lt;LINKID&gt; 4. L, n                                n = of parameter groups     1. L, 2         1. &lt;SPNAME<sub>1</sub>&gt;                Service parameter 1 name         2. &lt;SPVAL<sub>1</sub>&gt;                Service parameter 1 value     2. L, 2         1. &lt;SPNAME<sub>2</sub>&gt;                Service parameter 2 name         2. &lt;SPVAL<sub>2</sub>&gt;                Service parameter 2 value     .     .     n. L, 2         1. &lt;SPNAME<sub>n</sub>&gt;                Service parameter n name         2. &lt;SPVAL<sub>n</sub>&gt;                Service parameter n value 5. L, 2     1. &lt;SVCACK&gt;     2. L, p                                p = number of errors reported         1. L, 2             1. &lt;ERRCODE<sub>p</sub>&gt;             2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If p = 0, no errors were detected. If n = 0 no parameters are returned.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F22 Generic Service Completion Acknowledge (GSCA)	S,H<->E
<i>Description</i>	
The acknowledgement of generic Object Service Completion Information (i.e., S14F21).	
<i>Structure</i>	
<DATAACK>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F23 Multi-block Generic Service Data Inquire (GSDI)	S,H<->E, reply
<i>Description</i>	
If any of Object Service messages are larger than one block, then this transaction must precede that message.	
<i>Structure</i>	
L, 2 1. <DATAID> 2. <DATALENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F24 Multi-block Generic Service Data Grant (GSDG)	S,H<->E
<i>Description</i>	
Message to indicate if permission is granted to transmit a multi-block Object Service message.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F25 Get Service Name Request (GSNR)	S,H<->E, reply
<i>Description</i>	
This message is used to request to list the services of specified types of owned objects. This is an operation performed on an object type rather than on object instances. Wild characters “?” and “*” may be used as a filter for object types. Service provider is not required to support wild characters.	
<i>Structure</i>	
L, 2 1. <OBJSPEC> 2. L,n <span style="float: right;">n = # of object types</span> 1. <OBJTYP <sub>1</sub> > . . n. <OBJTYP <sub>n</sub> >	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no object specifier is provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F26 Get Service Name Data (GSND)	S,H<->E
<i>Description</i>	
This message contains a list of the services of the requested objects.	
<i>Structure</i>	
<pre> L, 2   1. L, n                                n = number of object types       1. L, 2           1. &lt;OBJTYP<sub>1</sub>&gt;           2. L, a                        a = number of attributes               1. &lt;SVCNAME<sub>1</sub>&gt;               .               .               a. &lt;SVCNAME<sub>a</sub>&gt;           .           .       n. L, 2           1. &lt;OBJTYP<sub>n</sub>&gt;           2. L, b                        b = number of attributes               1. &lt;SVCNAME<sub>1</sub>&gt;               .               .               b. &lt;SVCNAME<sub>b</sub>&gt;   2. L, 2       1. L, 2           1. &lt;OBJACK&gt;           2. L, p                        p = number of errors reported               1. L, 2                   1. &lt;ERRCODE<sub>1</sub>&gt;                   2. &lt;ERRTEXT<sub>1</sub>&gt;                   .                   .               p. L, 2                   1. &lt;ERRCODE<sub>p</sub>&gt;                   2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If p = 0, no errors were detected.	





<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F27 Get Service Parameter Name Request (GPNR)	S,H<->E, reply
<i>Description</i>	
This message is used to request to list the parameters of specified services of owned object. This is an operation performed on an object type rather than on object instances.	
<i>Structure</i>	
<pre>L, 3   1. &lt;OBJSPEC&gt;   2. &lt;OBJTYP&gt;   3. L,n                      n = # of interesting services       1. &lt;SVCNAME<sub>1</sub>&gt;       .       .       n. &lt;SVCNAME<sub>n</sub>&gt;</pre>	
<i>Exception</i>	
If OBJSPEC is a zero-length item, no object specifier is provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S14,F28 Get Service Parameter Name Data (GPND)	S,H<->E
<i>Description</i>	
This message contains a list of the service parameters of the requested services for the specified object.	
<i>Structure</i>	
<pre> L, 2   1. L, n                                n = number of services of interest       1. L, 2           1. &lt;SVCNAME<sub>1</sub>&gt;           2. L, a                        a = number of parameter names               1. &lt;SPNAME<sub>1</sub>&gt;               .               .               a. &lt;SPNAME<sub>a</sub>&gt;           .           .       n. L, 2           1. &lt;SVCNAME<sub>n</sub>&gt;           2. L, b                        b = number of parameter names               1. &lt;SPNAME<sub>1</sub>&gt;               .               .               b. &lt;SPNAME<sub>b</sub>&gt;   2. L, 2       1. &lt;OBJACK&gt;       2. L, p                            p = number of errors reported           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If p = 0, no errors were detected.	

10.19 *Stream 15 Recipe Management* — The functions in this stream are used requesting information and operations concerning recipes, recipe namespaces, and recipe executors. A recipe is an object that is transferred in sections, where a section consists of either recipe attributes, agent-specific dataset attributes, or the body of the recipe. An attribute is information concerning the recipe body, the recipe as a whole, or the application of the recipe. An attribute consists of an attribute name/attribute value pair.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F0 Abort Transaction (S15F0)	S,H<->E
<i>Description</i>	
Same form as S1,F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F1 Recipe Management Multi-block Inquire	S,H<->E,reply
<i>Description</i>	
This message requests permission to send a multi-block message based upon a maximum length of the total message.	
<i>Structure</i>	
L, 3 1. <DATAID> 2. <RCPSPEC> 3. <RMDATASIZE>	
<i>Exception</i>	
If RCPSPEC is zero-length, the multi-block message for which permission to send is requested does not contain a recipe.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F2 Recipe Management Multi-block Grant	S,H<->E
<i>Description</i>	
This message grants or denies permission to send a multi-block message.	
<i>Structure</i>	
<RMGRNT>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F3 Recipe Namespace Action Request	S,H<->E,reply
<i>Description</i>	
This message requests that a recipe namespace be created or deleted.	
<i>Structure</i>	
L, 2 1. <RMNSSPEC> 2. <RMNSCMD>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F4 Recipe Namespace Action Acknowledge	M,H<->E
<i>Description</i>	
This message is used to confirm whether the requested action was completed successfully or to provide error information otherwise.	
<i>Structure</i>	
L, 2 1. <RMACK> 2. L, p 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2 1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F5 Recipe Namespace Rename Request	S,H<->E,reply
<i>Description</i>	
A request is made for a recipe namespace to be renamed.	
<i>Structure</i>	
L, 2 1. <RMNSSPEC> 2. <RMNEWNS>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F6 Recipe Namespace Rename Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge or deny a request to rename a recipe namespace.	
<i>Structure</i>	
L, 2 1. <RMACK> 2. L, p 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2 1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F7 Recipe Space Request	S,H<->E,reply
<i>Description</i>	
This message requests the amount of recipe storage available in the storage of a recipe namespace or recipe executor, as indicated by its object specifier OBJSPEC.	
<i>Structure</i>	
<OBJSPEC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F8 Recipe Space Data	M,H<->E
<i>Description</i>	
This message contains the amount of storage available for recipes.	
<i>Structure</i>	
L, 2 1. <RMSPACE> 2. L, 2 1. <RMACK> 2. L, p 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2 1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F9 Recipe Status Request	S,H<->E,reply
<i>Description</i>	
This message is used to request the status of a recipe and the next available numeric version for that recipe class and name.	
<i>Structure</i>	
<RCPSPEC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F10 Recipe Status Data	M,H<->E
<i>Description</i>	
This message contains the protected status of the recipe and the next available version number for that recipe class and name.	
<i>Structure</i>	
<pre> L, 3   1. &lt;RCPSTAT&gt;   2. &lt;RCPVERS&gt;   3. L, 2       1. &lt;RMACK&gt;       2. L, p           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .       p. L, 2           1. &lt;ERRCODE<sub>p</sub>&gt;           2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
RCPVERS is a zero-length item if and only if the recipe does not exist. p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F11 Recipe Version Request	S,H<->E,reply
<i>Description</i>	
This message is used to request the best version of a recipe for the specified agent.	
<i>Structure</i>	
<pre> L, 4   1. &lt;RMNSSPEC&gt;   2. &lt;RCPCLASS&gt;   3. &lt;RCPNAME&gt;   4. &lt;AGENT&gt; </pre>	
<i>Exception</i>	
If item 2 is zero length, the recipe class PROCESS is indicated. If item 4 is a zero-length item, no agent is specified.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F12 Recipe Version Data	M,H<->E
<i>Description</i>	
This message contains the recommended version.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;AGENT&gt;</li> <li>2. &lt;RCPVERS&gt;</li> <li>3. L, 2 <ol style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>p. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ol> </li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
If AGENT is a zero-length item in the request, it shall also be a zero-length item in the reply. If it is not zero-length in the request, and it is of zero-length in the reply, then no qualifying recipe was found specific to that equipment. p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F13 Recipe Create Request	M,H<->E,reply
<i>Description</i>	
This message is used to create or modify a recipe body. If multi-block, it shall be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
L, 5 <ol style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;RCPUPDT&gt;</li> <li>3. &lt;RCPSPEC&gt;</li> <li>4. L, m <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;RCPATTRID<sub>1</sub>&gt;</li> <li>2. &lt;RCPATTRDATA<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>m. L, 2 <ol style="list-style-type: none"> <li>1. &lt;RCPATTRID<sub>m</sub>&gt;</li> <li>2. &lt;RCPATTRDATA<sub>m</sub>&gt;</li> </ol> </li> </ol> </li> <li>5. &lt;RCPBODY&gt;</li> </ol>	
<i>Exception</i>	
RCPBODY may be of zero length.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F14 Recipe Create Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge that a recipe has been created or updated with the body sent in the request.	
<i>Structure</i>	
<pre>L, 2   1. &lt;RMACK&gt;   2. L, 2       1. L, 2           1. &lt;ERRCODE<sub>1</sub>&gt;           2. &lt;ERRTEXT<sub>1</sub>&gt;       .       .       p. L, 2           1. &lt;ERRCODE<sub>p</sub>&gt;           2. &lt;ERRTEXT<sub>p</sub>&gt;</pre>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	



Stream,Function Name (Mnemonic)	Direction
S15,F15 Recipe Store Request	M,H<->E,reply
<i>Description</i>	
This message is used to send a recipe, or one or more recipe sections, to a recipe namespace. If multi-block, it shall be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
<pre> L, 4   1. &lt;DATAID&gt;   2. &lt;RCPSPEC&gt;   3. &lt;RCPSECCODE&gt;   4. L,q                                     (q = 1,2,3)       1. L,r                               (r = 0 or 2)           1. &lt;RCPSECNM&gt;           2. L,g                           (g = # generic attributes)               1. L,2                   1. &lt;RCPATTRID<sub>1</sub>&gt;                   2. &lt;RCPATTRDATA<sub>1</sub>&gt;               .               .           g. L,2               1. &lt;RCPATTRID<sub>g</sub>&gt;               2. &lt;RCPATTRDATA<sub>g</sub>&gt;       2. &lt;RCPBODY&gt;       3. L,m                               (m = # agent-specific datasets)           1. L,2               1. &lt;RCPSECNM<sub>1</sub>&gt;               2. L,a                   1. L,2                       1. &lt;RCPATTRID<sub>11</sub>&gt;                       2. &lt;RCPATTRDATA<sub>11</sub>&gt;                   .                   .               a. L,2                   1. &lt;RCPATTRID<sub>1a</sub>&gt;                   2. &lt;RCPATTRDATA<sub>1a</sub>&gt;               .               .           m. L,2               1. &lt;RCPSECNM<sub>m</sub>&gt;               2. L,b                   1. L,2                       1. &lt;RCPATTRID<sub>m1</sub>&gt;                       2. &lt;RCPATTRDATA<sub>m1</sub>&gt;                   .                   .               b. L,2                   1. &lt;RCPATTRID<sub>mb</sub>&gt;                   2. &lt;RCPATTRDATA<sub>mb</sub>&gt; </pre>	
<i>Exception</i>	
RCPBODY is a zero-length item when the body is omitted. If g = 0, no generic attributes are transferred and RCPBODY shall be a zero-length item. If m = 0, no agent-specific datasets are transferred.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F16 Recipe Store Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge that the specified recipe has been stored as requested or to indicate the error(s).	
<i>Structure</i>	
<pre> L, 2   1. &lt;RECPSECCODE&gt;   2. L, 2     1. &lt;RMACK&gt;     2. L, p       1. L, 2         1. &lt;ERRCODE<sub>1</sub>&gt;         2. &lt;ERRTEXT<sub>1</sub>&gt;       .       .       .     p. L, 2       1. &lt;ERRCODE<sub>p</sub>&gt;       2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F17 Recipe Retrieve Request	S,H<->E,reply
<i>Description</i>	
This message is used to get a recipe, or one or more recipe sections, from a recipe namespace.	
<i>Structure</i>	
<pre> L, 2   1. &lt;RCPSPEC&gt;   2. &lt;RCPSECCODE&gt; </pre>	
<i>Exception</i>	
None	

Stream,Function Name (Mnemonic)	Direction
S15,F18 Recipe Retrieve Data	M,H<->E
<i>Description</i>	
This message is used to acknowledge that the specified recipe, or recipe sections, have been set as requested, or to indicate the error(s).	
<i>Structure</i>	
<pre> L, 2   1. L,q                                     (q = 1,2,3)     1. L,r                                   (r = 0 or 2)       1. &lt;RCPSECNM&gt;       2. L,g                               (g = # generic attributes)         1. L,2           1. &lt;RCPATTRID<sub>1</sub>&gt;           2. &lt;RCPATTRDATA<sub>1</sub>&gt;           .           .         g. L,2           1. &lt;RCPATTRID<sub>g</sub>&gt;           2. &lt;RCPATTRDATA<sub>g</sub>&gt;       2. &lt;RCPBODY&gt;       3. L,m                               (m = # agent-specific datasets)         1. L,2           1. &lt;RCPSECNM<sub>1</sub>&gt;           2. L,a             1. L,2               1. &lt;RCPATTRID<sub>11</sub>&gt;               2. &lt;RCPATTRDATA<sub>11</sub>&gt;               .               .             a. L,2               1. &lt;RCPATTRID<sub>1a</sub>&gt;               2. &lt;RCPATTRDATA<sub>1a</sub>&gt;             .             .           m. L,2             1. &lt;RCPSECNM<sub>m</sub>&gt;             2. L,b               1. L,2                 1. &lt;RCPATTRID<sub>m1</sub>&gt;                 2. &lt;RCPATTRDATA<sub>m1</sub>&gt;                 .                 .               b. L,2                 1. &lt;RCPATTRID<sub>mb</sub>&gt;                 2. &lt;RCPATTRDATA<sub>mb</sub>&gt;             2. L,2               1. &lt;RMACK&gt;               2. L,p                 1. L,2                   1. &lt;ERROCODE<sub>1</sub>&gt;                   2. &lt;ERRTEXT<sub>1</sub>&gt;                 .           .         </pre>	

<p>.</p> <p>p. L, 2</p> <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ol>
<i>Exception</i>
If r = 0, no generic attributes are transferred and RCPBODY shall be a zero-length item. If m = 0, no agent-specific datasets are transferred. p = 0 if and only if RMACK indicates no errors.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F19 Recipe Rename Request	S,H<->E,reply
<i>Description</i>	
This message is used to request that a recipe be copied to, or renamed to, a recipe with a new identifier.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;RCPSPEC&gt;</li> <li>2. &lt;RCPRENAME&gt;</li> <li>3. &lt;RCPNEWID&gt;</li> </ol>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F20 Recipe Rename Acknowledge	M,H<->E
<i>Description</i>	
This message acknowledges the request to copy or rename a recipe and indicates whether the action was successfully performed or errors that occurred.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>p. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F21 Recipe Action Request	M,H<->E,reply
<i>Description</i>	
This message is used to acknowledge the request to perform an action in one or more recipes within a namespace.	
<i>Structure</i>	
L, 6 <ol style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;RCPCMD&gt;</li> <li>3. &lt;RMNSSPEC&gt;</li> <li>4. &lt;OPID&gt;</li> <li>5. &lt;AGENT&gt;</li> <li>6. L, n <ol style="list-style-type: none"> <li>1. &lt;RCPID<sub>1</sub>&gt;</li> </ol> </li> </ol>	

. . n. <RCPID <sub>n</sub> >
<i>Exception</i>
AGENT may be a zero-length item except for requests for certify, de-certify, download, and upload.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F22 Recipe Action Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge the request to originate a new recipe.	
<i>Structure</i>	
L, 4 1. <AGENT> 2. <LINKID> 3. <RCPCMD> 4. L, 2 1. <RMACK> 2. L, p 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2 1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
LINKID is zero if and only if all requested actions have been completed. p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F23 Recipe Descriptor Request	M,H<->E,reply
<i>Description</i>	
This message is used to request the descriptors of a list of the specified recipes. If multi-block, it must be preceded by the S15,F1/F2 inquire/grant transaction. OBJSPEC is the object specifier of either a recipe namespace of a recipe executor.	
<i>Structure</i>	
L, 3 1. <DATAID> 2. <OBJSPEC> 3. L, n 1. <RCPID <sub>1</sub> > . . n. <RCPID <sub>n</sub> >	
<i>Exception</i>	
None	

Stream,Function Name (Mnemonic)	Direction
S15,F24 Recipe Descriptor Data	M,H<->E
<i>Description</i>	
This message returns the requested descriptors in the same order as requested.	
<i>Structure</i>	
<pre> L,2   1. L,n                      (n = number of recipes from request)     1. L,a                    (descriptors for recipe #1)       1. L,r                  (r = 0 or 3) (1st component descriptor)         1. &lt;RCPDESCNM<sub>11</sub>&gt;         2. &lt;RCPDESCTIME<sub>11</sub>&gt;         3. &lt;RCPDESCLTH<sub>11</sub>&gt;         .         .       a. L,r                  (r = 0 or 3)         1. &lt;RCPDESCNM<sub>1a</sub>&gt;         2. &lt;RCPDESCTIME<sub>1a</sub>&gt;         3. &lt;RCPDESCLTH<sub>1a</sub>&gt;         .         .     n. L,b                    (descriptors for recipe #n)       1. L,r                  (r = 0 or 3) (1st component descriptor)         1. &lt;RCPDESCNM<sub>n1</sub>&gt;         2. &lt;RCPDESCTIME<sub>n1</sub>&gt;         3. &lt;RCPDESCLTH<sub>n1</sub>&gt;         .         .       b. L,r                  (r = 0 or 3)         1. &lt;RCPDESCNM<sub>nb</sub>&gt;         2. &lt;RCPDESCTIME<sub>nb</sub>&gt;         3. &lt;RCPDESCLTH<sub>nb</sub>&gt;   2. L,2     1. &lt;RMACK&gt;     2. L,p       1. L,2         1. &lt;ERRCODE<sub>1</sub>&gt;         2. &lt;ERRTEXT<sub>1</sub>&gt;         .         .       p. L,2         1. &lt;ERRCODE<sub>p</sub>&gt;         2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
A zero-length recipe descriptor (r = 0) means that the specified recipe does not exist (could not be located). p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F25 Recipe Parameter Update Request	M,H<->E,reply
<i>Description</i>	
This message is used to update the variable parameter definitions for a specific agent. If multi-block, it must be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
L, 4 <ol style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;RMNSSPEC&gt;</li> <li>3. &lt;AGENT&gt;</li> <li>4. L, n <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;RCPPARNM<sub>1</sub>&gt;</li> <li>2. &lt;RCPPARVAL<sub>1</sub>&gt;</li> <li>3. &lt;RCPPARRULE<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 3 <ol style="list-style-type: none"> <li>1. &lt;RCPPARNM<sub>n</sub>&gt;</li> <li>2. &lt;RCPPARVAL<sub>n</sub>&gt;</li> <li>3. &lt;RCPPARRULE<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F26 Recipe Parameter Update Acknowledge	M, H<->E
<i>Description</i>	
This message indicates the successful performance of the request or otherwise indicates the nature of error(s) that occurred.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>p. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F27 Recipe Download Request	M,H->E,reply
<i>Description</i>	
This message is used to send a recipe to a recipe executor. If multi-block, it shall be preceded by the S15,F1/S15,F2 inquire/grant transaction.	
<i>Structure</i>	
L, 5 1. <DATAID> 2. <RCPOWCODE> 3. <RCPSPEC> 4. L, m 1. L, 2 1. <RCPATTRID <sub>1</sub> > 2. <RCPATTRDATA <sub>1</sub> > . . m. L, 2 1. <RCPATTRID <sub>m</sub> > 2. <RCPATTRDATA <sub>m</sub> > 5. <RCPBODY>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F28 Recipe Download Acknowledge	M,H<-E
<i>Description</i>	
This message is used to acknowledge that a recipe has been received by the recipe executor. If the recipe was successfully verified, the results are returned to the sender. RCPID contains the identifier of a derived object form recipe if created during verification.	
<i>Structure</i>	
L, 3 1. <RCPID> 2. L, n (n = # of attributes) 1. L, 2 1. <RCPATTRID <sub>1</sub> > 2. <RCPATTRDATA <sub>1</sub> > . . n. L, 2 1. <RCPATTRID <sub>n</sub> > 2. <RCPATTRDATA <sub>n</sub> > 3. L, 2 1. <RMACK> 2. L, p 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2 1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
If item is a zero length item, no derived object form recipe was originated. n = 0 if and only if the recipe was not verified or failed verification. p = 0 if and only if RMACK indicates no errors.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F29 Recipe Verify Request	M,H->E,reply
<i>Description</i>	
This message is used to request verification of one or more recipes by a recipe executor. If multi-block, it shall be preceded by the S15F1,F2 inquire/grant transaction. The operation identifier OPID, used where multiple verification requests may be outstanding, may be zero if no further verifications will be requested before all current verification requests are completed by the recipe executor. Otherwise, OPID is generated to be unique for the requestor. RESPEC is the object specifier for the recipe executor.	
<i>Structure</i>	
L, 4 1. <DATAID> 2. <OPID> 3. <RESPEC> 4. L, m 1. <RCPID <sub>1</sub> > . . m. <RCPID <sub>m</sub> >	
<i>Exception</i>	
If RESPEC is a zero length item, the target is the recipient of the message.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F30 Recipe Verify Acknowledge	M,H<-E
<i>Description</i>	
This message is used to acknowledge the request to verify one or more recipes. If a single recipe verification was requested and the recipe was successfully verified, the results are returned to the sender in this message, and RCPID contains the identifier of a derived object form recipe if created during verification. If multiple recipe verifications were requested, then LINKID shall be non-zero.	
<i>Structure</i>	
<pre> L, 5   1. &lt;OPID&gt;   2. &lt;LINKID&gt;   3. &lt;RCPID&gt;   4. L, n                                (n = # attributes)       1. L, 2           1. &lt;RCPATTRID<sub>1</sub>&gt;           2. &lt;RCPATTRDATA<sub>1</sub>&gt;       .       .       n. L, 2           1. &lt;RCPATTRID<sub>n</sub>&gt;           2. &lt;RCPATTRDATA<sub>n</sub>&gt;   5. L, 2       1. &lt;RMACK&gt;       2. L, p           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
LINKID is zero if and only if a single recipe verification was requested and has been completed. If item 3 is zero length item, no derived object form recipe was originated. n = 0 if and only if the recipe was not verified or failed verification. p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F31 Recipe Upload Request	S,H->E,reply
<i>Description</i>	
This message is used to request an execution recipe from a recipe executor.	
<i>Structure</i>	
<RCPSPEC>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F32 Recipe Upload Data	M,H<-E
<i>Description</i>	
This message is used to send an execution recipe from a recipe executor.	
<i>Structure</i>	
<pre> L, 4   1. &lt;RCPSPEC&gt;   2. L,m                      (m = # attributes)       1. L,2           1. &lt;RCPATTRID<sub>1</sub>&gt;           2. &lt;RCPATTRDATA<sub>1</sub>&gt;       .       .       m. L,2           1. &lt;RCPATTRID<sub>m</sub>&gt;           2. &lt;RCPATTRDATA<sub>m</sub>&gt;   3. &lt;RCPBODY&gt;   4. L,2       1. &lt;RMACK&gt;       2. L,p           1. L,2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L,2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F33 Recipe Select Request	M,H->E,reply
<i>Description</i>	
This message is used to request the selection of one or more execution recipes. If multi-block, it shall be preceded by the S15,F1/S15,F2 inquire/grant transaction.	
<i>Structure</i>	
<pre> L, 3   1. &lt;DATAID&gt;   2. &lt;RESPEC&gt;   3. L,r                                     (r = # selections)       1. L,2           1. &lt;RCPID<sub>1</sub>&gt;                       (1st recipe selection)           2. L,p                             (p = # parameter settings for 1st recipe)               1. L,2                   1. &lt;RCPPARNM<sub>11</sub>&gt;                   2. &lt;RCPPARVAL<sub>11</sub>&gt;               .               .               p. L,2                   1. &lt;RCPPARNM<sub>1p</sub>&gt;                   2. &lt;RCPPARVAL<sub>1p</sub>&gt;           .           .       r. L,2           1. &lt;RCPID<sub>r</sub>&gt;                       (rth recipe selection)           2. L,s                             (s = # parameter settings for rth recipe)               1. L,2                   1. &lt;RCPPARNM<sub>r1</sub>&gt;                   2. &lt;RCPPARVAL<sub>r1</sub>&gt;               .               .               s. L,2                   1. &lt;RCPPARNM<sub>rs</sub>&gt;                   2. &lt;RCPPARVAL<sub>rs</sub>&gt; </pre>	
<i>Exception</i>	
If the list of parameter settings for a recipe selection is of zero length, then no parameter settings are specified for the corresponding recipe.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F34 Recipe Select Acknowledge	M,H<-E
<i>Description</i>	
This message is used to acknowledge the request for recipe selection.	
<i>Structure</i>	
L, 2 1. <RMACK> 2. L, p 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . p. L, 2 1. <ERRCODE <sub>p</sub> > 2. <ERRTEXT <sub>p</sub> >	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F35 Recipe Delete Request	M,H->E,reply
<i>Description</i>	
This message is used to request that one or more recipes be deleted or deselected. If multi-block, it shall be preceded by the S15,F1/S15,F2 inquire/grant transaction.	
<i>Structure</i>	
L, 4 1. <DATAID> 2. <RESPEC> 3. <RCPDEL> 4. L, n (n = # recipes deselected) 1. <RCPID <sub>1</sub> > . . n. <RCPID <sub>n</sub> >	
<i>Exception</i>	
If n = 0 and recipes are to be deselected (RCPDEL = 1), then all currently-selected recipes are indicated.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F36 Recipe Delete Acknowledge	M,H<-E
<i>Description</i>	
This message is used to acknowledge the request that recipes be deleted or deselected.	
<i>Structure</i>	
L, 2 <ul style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ul style="list-style-type: none"> <li>1. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ul> </li> <li>.</li> <li>.</li> </ul> </li> <li>p. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ul> </li> </ul>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F37 DRNS Segment Approve Action Request	S,H<->E,reply
<i>Description</i>	
This message is sent by a distributed recipe namespace manager to an attached distributed recipe namespace segment to approve an action previously requested by the segment. If multi-block, it shall be preceded by the S15,F1/S15,F2 inquire/grant transaction.	
<i>Structure</i>	
L, 6 <ul style="list-style-type: none"> <li>1. &lt;RMSEGSPEC&gt;</li> <li>2. &lt;OBJTOKEN&gt;</li> <li>3. &lt;RMGRNT&gt;</li> <li>4. &lt;OPID&gt;</li> <li>5. &lt;RCPID&gt;</li> <li>6. &lt;RMCHGTYPE&gt;</li> </ul>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F38 DRNS Segment Approve Action Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge or deny the approve action request.	
<i>Structure</i>	
L, 2 <ul style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ul style="list-style-type: none"> <li>1. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ul> </li> <li>.</li> <li>.</li> </ul> </li> <li>p. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ul> </li> </ul>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F39 DRNS Recorder Segment Request	M,H<->E,reply
<i>Description</i>	
This message is used by the distributed recipe namespace manager to request that an attached recorder create or delete a segment specifier record. If multi-block, it shall be preceded by the S15,F1/S15,F2 inquire/grant transaction.	
<i>Structure</i>	
L, 5 <ul style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;RMNSCMD&gt;</li> <li>3. &lt;RMRECSPEC&gt;</li> <li>4. &lt;RMSEGSPEC&gt;</li> <li>5. &lt;OBJTOKEN&gt;</li> </ul>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F40 DRNS Recorder Segment Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge the request to add or delete a segment specifier record.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> </ol> </li> <li>p. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F41 DRNS Recorder Modify Request	M,H<->E,reply
<i>Description</i>	
This message is used by a distributed recipe namespace manager to a recorder to store or delete a change request record. If multi-block, it shall be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
L, 5 <ol style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;RMRECSPEC&gt;</li> <li>3. &lt;OBJTOKEN&gt;</li> <li>4. &lt;RMNSCMD&gt;</li> <li>5. L, c (c = 1 or 7) <ol style="list-style-type: none"> <li>1. &lt;RCPID&gt;</li> <li>2. &lt;RCPNEWID&gt;</li> <li>3. &lt;RMSEGSPEC&gt;</li> <li>4. &lt;RMCHGTYPE&gt;</li> <li>5. &lt;OPID&gt;</li> <li>6. &lt;TIMESTAMP&gt;</li> <li>7. &lt;RMREQUESTOR&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
If RMNSCMD = create, then c = 7, otherwise c = 1.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F42 DRNS Recorder Modify Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge a request to store or delete a change request.	
<i>Structure</i>	
L, 2 <ul style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, p <ul style="list-style-type: none"> <li>1. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> </ul> </li> <li>p. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ul> </li> </ul> </li> </ul>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F43 DRNS Get Change Request	M,H<->E,reply
<i>Description</i>	
This message is used to request a distributed recipe namespace recorder or manager to return change requests records for a specific recipe or assigned to a specific segment. If multi-block, it shall be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
L, 3 <ul style="list-style-type: none"> <li>1. &lt;DATAID&gt;</li> <li>2. &lt;OBJSPEC&gt;</li> <li>3. &lt;TARGETSPEC&gt;</li> </ul>	
<i>Exception</i>	
If TARGETSPEC is omitted, OBJSPEC identifies a recipe.	

Stream,Function Name (Mnemonic)	Direction
S15,F44 DRNS Get Change Request Data	M,H<->E
<i>Description</i>	
This message is used to return the specified change request records.	
<i>Structure</i>	
<pre> L, 2   1. L, n                                n = # change requests       1. L, 7           1. &lt;RCPID<sub>1</sub>&gt;           2. &lt;RCPNEWID<sub>1</sub>&gt;           3. &lt;RMSEGSPEC<sub>1</sub>&gt;           4. &lt;RMCHGTYPE<sub>1</sub>&gt;           5. &lt;OPID<sub>1</sub>&gt;           6. &lt;TIMESTAMP<sub>1</sub>&gt;           7. &lt;RMREQUESTOR<sub>1</sub>&gt;       .       .       n. L, 7           1. &lt;RCPID<sub>n</sub>&gt;           2. &lt;RCPNEWID<sub>n</sub>&gt;           3. &lt;RMSEGSPEC<sub>n</sub>&gt;           4. &lt;RMCHGTYPE<sub>n</sub>&gt;           5. &lt;OPID<sub>n</sub>&gt;           6. &lt;TIMESTAMP<sub>n</sub>&gt;           7. &lt;RMREQUESTOR<sub>n</sub>&gt;   2. L, 2       1. &lt;RMACK&gt;       2. L, p           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           p. L, 2               1. &lt;ERRCODE<sub>p</sub>&gt;               2. &lt;ERRTEXT<sub>p</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, no change records were found matching the specification. p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F45 DRNS Manager Segment Change Approval Request	M,H<->E,reply
<i>Description</i>	
This message is sent to a distributed recipe namespace manager by an attached distributed recipe namespace segment to request approval for a specific type of change to a recipe. If multi-block, it shall be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
L, 4 1. <DATAID> 2. <RCPSPEC> 3. <RCPNEWID> 4. <RMCHGTYPE>	
<i>Exception</i>	
RCPNEWID is a zero-length item except where RMCHGTYPE specifies a copy or rename change.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F46 DRNS Manager Segment Approval Acknowledge	S,H<->E
<i>Description</i>	
This message is used to acknowledge the request to change a recipe.	
<i>Structure</i>	
L, 3 1. <RMCHGTYPE> 2. <RMGRNT> 3. <OPID>	
<i>Exception</i>	
OPID is zero if and only if RMGRNT indicates the change is denied.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F47 DRNS Manager Rebuild Request	M,H<->E,reply
<i>Description</i>	
This message requests a distributed recipe namespace manager specified in OBJSPEC to rebuild a distributed recipe namespace. Either a distributed recipe namespace recorder or a list of distributed recipe namespace segment specifiers shall be provided. If multi-block, it shall be preceded by the S15,F1/F2 inquire/grant transaction.	
<i>Structure</i>	
L, 5 1. <DATAID> 2. <OBJSPEC> 3. <RMNSSPEC> 4. <RMRECSPEC> 5. L, n 1. <RMSEGSPEC <sub>1</sub> > . . n. <RMSEGSPEC <sub>n</sub> >	
<i>Exception</i>	
If RMRECSPEC is a non-zero length item, then n is zero. If RMRECSPEC is a zero length item, then n is non-zero.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F48 DRNS Manager Rebuild Acknowledge	M,H<->E
<i>Description</i>	
This message is used to acknowledge the request to rebuild a distributed recipe namespace.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, P <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> </ol> </li> <li>p. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>p</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>p</sub>&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
p = 0 if and only if RMACK indicates no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F49 Large Recipe Download Request (LRDR)	S,H->E,reply
<i>Description</i>	
This is a request by the host for the equipment to request the download of a recipe via the Stream 13 Data Set Transfer protocol. The Data Set name, DSNAME, is the text string identifier of the recipe, RCPSPEC. The Data Set is subsequently transferred as a Stream with the following internal SECSII structured data:	
L, 4 <ol style="list-style-type: none"> <li>1. &lt;RCPSPEC&gt;</li> <li>2. &lt;DATAID&gt;</li> <li>3. L, m (m = # of attributes) <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;RCPATTRID<sub>1</sub>&gt;</li> <li>2. &lt;RCPATTRDATA<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> </ol> </li> <li>m. L, 2 <ol style="list-style-type: none"> <li>1. &lt;RCPATTRID<sub>m</sub>&gt;</li> <li>2. &lt;RCPATTRDATA<sub>m</sub>&gt;</li> </ol> </li> <li>4. &lt;RCPBODY&gt;</li> </ol>	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;DSNAME&gt;</li> <li>2. &lt;RCPOWCODE&gt;</li> </ol>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F50 Large Recipe Download Acknowledge (LRDA)	S,H<-E
<i>Description</i>	
Acknowledge or error. A returned status of “accepted” means only that the message is understood. Upon completion of the large recipe download request (Stream 13 Data Set transfer scenario) the equipment initiates a separate verification transaction (S15,F53/S15,F54) that provides the result of the verification.	
<i>Structure</i>	
<ACKC15>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F51 Large Recipe Upload Request (LRUR)	S,H->E,reply
<i>Description</i>	
This is a request by the host for the equipment to request the host to upload a recipe via the Stream 13 Data Set Transfer protocol. The Data Set name, DSNAME, is the text string identifier of the recipe, RCPSPEC. The Data Set is subsequently transferred as a Stream with the following internal SECSII structured data:	
L, 3 1. <RCPSPEC> 2. L, m (m = # of attributes) 1. L, 2 1. <RCPATTRID <sub>1</sub> > 2. <RCPATTRDATA <sub>1</sub> > . . m. L, 2 1. <RCPATTRID <sub>m</sub> > 2. <RCPATTRDATA <sub>m</sub> > 3. <RCPBODY>	
<i>Structure</i>	
<DSNAME>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F52 Large Recipe Upload Acknowledge (LRUA)	S,H<-E
<i>Description</i>	
Acknowledge or error. A returned status of “accepted” means only that the message is understood. The completion of the request is signaled by an event report.	
<i>Structure</i>	
<ACKC15>	
<i>Exception</i>	
It is possible to use the ACKC15 code “command will be performed with completion signaled later” for this message.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F53 Recipe Verification Send (RVS)	M,H<-E,reply
<i>Description</i>	
This message indicates to the host that a large recipe that was transferred via Stream 13 Data Set Transfer Protocol had been received and checked by the equipment. RCPID contains the identifier of a derived object form recipe if created during verification. The result of the check is specified by the list of errors. An empty error list indicates no errors were found in the recipe. The equipment is responsible for sending a single copy of this message to the host after any reception of a recipe through S15,F49.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;RCPSPEC&gt;</li> <li>2. &lt;RCPID&gt;</li> <li>3. L, 2 <ol style="list-style-type: none"> <li>1. &lt;RMACK&gt;</li> <li>2. L, n (n = number of errors being reported) <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> <li>.</li> </ol> </li> <li>n. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>n</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
n = 0 if and only if RMACK indicates no error. If RCPSEPC is a zero length item, then the recipe was not verified or failed verification. If RCPID is zero length, then no derived object form recipe was originated.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S15,F54 Recipe Verification Acknowledge (RVA)	S,H->E
<i>Description</i>	
Reply by host to equipment providing response to Recipe Verification Send (RVS).	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

10.20 *Stream 16 Processing Management* — This stream provides protocol for a set of messages that enable the control of material processing at equipment and equipment resources. Control is implemented by supporting two job types; the control job and the process job. A process job is a single unit of work that ensures that the appropriate processing is applied to a particular material by a processing resource. The Process Job provides a widely applicable supervisory control capability for automated processing of material in equipment, irrespective of the particular process being used. The Process Job creates a transient link between the three elements of the manufacturing process; the first is the material to be processed. The second is the equipment on which the process will occur. The third is the process specification, a Process Recipe. When a Process Job has completed, it ceases to exist; its Process Job ID is no longer valid. The control job is used to group a set of related process jobs. The group is logically related from the host's viewpoint. For instance; if a carrier contains multiple lots, then the process jobs for each lot (in the carrier) could be included in the control job specification. Control jobs also provide mechanisms for specifying the destination for processed material.



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F0 Abort Transaction (S16F0)	S,H<->E
<i>Description</i>	
Same form as S1F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F1 Multi-block Process Job Data Inquire (PRJI)	S,H->E,reply
<i>Description</i>	
If any of Processing Management messages are larger than one block, then this transaction must precede that message.	
<i>Structure</i>	
L, 2 1. <DATAID> 2. <DATALENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F2 Multi-block Process Job Data Grant (PRJG)	S,H<-E
<i>Description</i>	
Message to indicate if permission is granted to transmit a multi-block Job Data message.	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F3 Process Job Create Request (PRJCR)	M,H->E,reply
<i>Description</i>	
The purpose of this message is to request material to be processed on a Process Module.	
<i>Structure</i>	
<pre> L, 5 1. &lt;DATAID&gt; 2. &lt;MF&gt; 3. L, n     1. &lt;MID<sub>1</sub>&gt;     .     .     n. &lt;MID<sub>n</sub>&gt; 4. L, 3     1. &lt;PRRECIPEMETHOD&gt;     2. &lt;RCPSPEC&gt;     3. L, m                                (m = {c, 2})         1. L, 2             1. &lt;RCPPARNM<sub>1</sub>&gt;             2. &lt;RCPPARVAL<sub>1</sub>&gt;         .         .         m. L, 2             1. &lt;RCPPARNM<sub>m</sub>&gt;             2. &lt;RCPPARVAL<sub>m</sub>&gt; 5. &lt;PRPROCESSSTART&gt; </pre>	
<i>Exception</i>	
For the m length list m = 0 may be allowed value depending on the value of PRRECIPEMETHOD.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F4 Process Job Create Acknowledge (PRJCA)	S,H<-E
<i>Description</i>	
Acknowledge or report error in the creation of a Process Job.	
<i>Structure</i>	
L, 2 1. <PRJOBID> 2. L, 2 1. <ACKA> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
This list may be zero length, generally the case when ACKA indicates success. When ACKA indicates a create failure, the equipment may supply one or more ERRCODEs.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F5 Process Job Command Request (PRJCMDR)	M,H->E,reply
<i>Description</i>	
Send a job control command to a processing job.	
<i>Structure</i>	
L, 4 1. <DATAID> 2. <PRJOBID> 3. <PRCMDNAME> 4. L, n 1. L, 2 1. <CPNAME <sub>1</sub> > 2. <CPVAL <sub>1</sub> > . . n. L, 2 1. <CPNAME <sub>n</sub> > 2. <CPVAL <sub>n</sub> >	
<i>Exception</i>	
The CPNAME, CPVAL pairs are command parameter identifiers and values; n = 0 is valid for some commands (PRCMDNAME).	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F6 Process Job Command Acknowledge (PRJCMDA)	S,H<-E
<i>Description</i>	
The processing service sends its confirmation for receipt of a command request.	
<i>Structure</i>	
L, 2 1. <PRJOBID> 2. L, 2 1. <ACKA> 2. L, n (n = {0,n}) 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
This list n may be zero length.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F7 Process Job Alert Notify (PRJA)	S,H<-E,[reply]
<i>Description</i>	
The processing service may notify the controlling entity of important events. The Process Job Milestones only assume small number of different values. However, the conditions under which a process job meets one of these milestones may vary. For instance, a Job may reach Job Complete because the Process was Aborted. By using item 4, the status of the Alert (PRJOBMILESTONE) can be indicated. See the list of Error Codes for Processing in Data Item Dictionary.	
<i>Structure</i>	
L, 4 <ul style="list-style-type: none"> <li>1. &lt;TIMESTAMP&gt;</li> <li>2. &lt;PRJOBID&gt;</li> <li>3. &lt;PRJOBMILESTONE&gt;</li> <li>4. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ACKA&gt;</li> <li>2. L, n (n = {0,n}) <ul style="list-style-type: none"> <li>1. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ul> </li> <li>.</li> <li>.</li> </ul> </li> <li>n. L, 2 <ul style="list-style-type: none"> <li>1. &lt;ERRCODE<sub>n</sub>&gt;</li> <li>2. &lt;ERRTEXT<sub>n</sub>&gt;</li> </ul> </li> </ul> </li> </ul>	
<i>Exception</i>	
The list n may be zero length.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F8 Process Job Alert Confirm (PRJAC)	S,H->E
<i>Description</i>	
Host confirms receipt of Process Job Alert message from the equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F9 Process Job Event Notify (PRJE)	S,H<-E,[reply]
<i>Description</i>	
Send Processing Job related event to the controlling entity.	
<i>Structure</i>	
L, 4 <ul style="list-style-type: none"> <li>1. &lt;PREVENTID&gt;</li> <li>2. &lt;TIMESTAMP&gt;</li> <li>3. &lt;PRJOBID&gt;</li> <li>4. L, n <ul style="list-style-type: none"> <li>1. L, 2 <ul style="list-style-type: none"> <li>1. &lt;VID<sub>1</sub>&gt;</li> <li>2. &lt;V<sub>1</sub>&gt;</li> </ul> </li> <li>.</li> <li>.</li> <li>n. L, 2 <ul style="list-style-type: none"> <li>1. &lt;VID<sub>n</sub>&gt;</li> <li>2. &lt;V<sub>n</sub>&gt;</li> </ul> </li> </ul> </li> </ul>	
<i>Exception</i>	
The VID, V pairs are variable data identifiers and values; exceptions n = 0 is valid for some events (PREVENTID).	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F10 Process Job Event Confirm (PRJEC)	S,H->E
<i>Description</i>	
Host confirms receipt of S16,F9 message to equipment.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

Stream,Function Name (Mnemonic)	Direction
S16,F11 PRJobCreateEnh	M,H->E,reply
<i>Description</i>	
Request equipment to create a Process Job with the given PRJOBID. If multi-block, this message must be preceded by the S16,F1/F2 transaction.	
<i>Structure</i>	
<pre> L, 7 1.  &lt;DATAID&gt; 2.  &lt;PRJOBID&gt; 3.  &lt;MF&gt; 4a. L,n                                [MF = carrier, n = # of carriers]     1. L,2         1. &lt;CARRIERID<sub>1</sub>&gt;         2. L,j                        [j = # of slots, may be implemented as an array]             1. &lt;SLOTID<sub>1</sub>&gt;             2. &lt;SLOTID<sub>2</sub>&gt;             .             .             j. &lt;SLOTID<sub>j</sub>&gt;         .     n. L,2         1. &lt;CARRIERID<sub>n</sub>&gt;         2. L,j                        [j = # of slots, may be implemented as an array]             1. &lt;SLOTID<sub>1</sub>&gt;             2. &lt;SLOTID<sub>2</sub>&gt;             .             .             j. &lt;SLOTID<sub>j</sub>&gt; 4b. L,n                                [MF = substrate]     1. &lt;MID<sub>1</sub>&gt;     .     .     n. &lt;MID<sub>n</sub>&gt; 5.  L,3     1. &lt;PRRECIPEMETHOD&gt;     2. &lt;RCPSPEC&gt;     3. L,m                            [m = # recipe parameters]         1. L,2             1. &lt;RCPPARM<sub>1</sub>&gt;             2. &lt;RCPPARVAL<sub>1</sub>&gt;         .         m. L,2             1. &lt;RCPPARM<sub>m</sub>&gt;             2. &lt;RCPPARVAL<sub>m</sub>&gt; 6.  &lt;PRPROCESSSTART&gt; 7.  &lt;PRPAUSEEVENT&gt; </pre>	
<i>Exception</i>	
The list for specifying material (item 4a and 4b) is empty (L,0 instead of L,n), when no material is specified for the process job. The form of data item 4 (a or b) depends on the value in MF.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F12 PRJobCreateEnh Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the request and reports any errors in the creation of a process job.	
<i>Structure</i>	
L, 2 1. <PRJOBID> 2. L, 2 1. <ACKA> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F13 PRJobDuplicateCreate	M,H->E,reply
<i>Description</i>	
Not used.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F14 PRJobDuplicateCreate Acknowledge	S,H<-E
<i>Description</i>	
Not used.	
<i>Structure</i>	
<i>Exception</i>	

Stream,Function Name (Mnemonic)	Direction
S16,F15 PRJobMultiCreate	M,H->E,reply
<i>Description</i>	
Use this single message to Create Multiple Process Jobs, each of which may be unique in its association of material to process recipe. If multi-block, this message must be preceded by the S16,F1/F2 transaction.	
<i>Structure</i>	
<pre> L,2   1. &lt;DATAID&gt;   2. L,p                [p = # of process jobs being created]     1. L,6       1. &lt;PRJOBID<sub>1</sub>&gt;       2. &lt;MF<sub>1</sub>&gt;       3a. L,n            [MF = carrier, n = # of carriers]         1. L,2           1. &lt;CARRIERID<sub>1</sub>&gt;           2. L,j        [j = # of slots, may be implemented as an array]             1. &lt;SLOTID<sub>1</sub>&gt;             2. &lt;SLOTID<sub>2</sub>&gt;             .             .             j. &lt;SLOTID<sub>j</sub>&gt;           .           .         n. L,2           1. &lt;CARRIERID<sub>n</sub>&gt;           2. L,j        [j = # of slots, may be implemented as an array]             1. &lt;SLOTID<sub>1</sub>&gt;             2. &lt;SLOTID<sub>2</sub>&gt;             .             .             j. &lt;SLOTID<sub>j</sub>&gt;       3b. L,n            [MF = substrate, n = # of MID]         1. &lt;MID<sub>1</sub>&gt;         .         .         n. &lt;MID<sub>n</sub>&gt;       4. L,3         1. &lt;PRRECIPEMETHOD<sub>1</sub>&gt;         2. &lt;RCPSPEC<sub>1</sub>&gt;         3. L,m          [m = # recipe parameters]           1. L,2             1. &lt;RCPPARNM<sub>1</sub>&gt;             2. &lt;RCPPARVAL<sub>1</sub>&gt;             .             .           m. L,2             1. &lt;RCPPARNM<sub>m</sub>&gt;             2. &lt;RCPPARVAL<sub>m</sub>&gt;         5. &lt;PRPROCESSSTART<sub>1</sub>&gt;         6. &lt;PRPAUSEEVENT<sub>1</sub>&gt;         .         . </pre>	

```

p. L,6
  1. <PRJOBIDp>
  2. <MFp>
  3a. L,n [MF = carrier, n = # of carriers]
    1. L,2
      1. <CARRIERID1>
      2. L,j [j = # of slots, may be implemented as an array]
        1. <SLOTID1>
        2. <SLOTID2>
        .
        .
        j. <SLOTIDj>
      .
    n. L,2
      1. <CARRIERIDn>
      2. L,j [j = # of slots, may be implemented as an array]
        1. <SLOTID1>
        2. <SLOTID2>
        .
        .
        j. <SLOTIDj>
  3b. L,n [MF = substrate, n = # of MID]
    1. <MID1>
    .
    .
    n. <MIDn>
  4. L,3
    1. <PRRECIPEMETHODp>
    2. <RCPSPECp>
    3. L,m [m = # recipe parameters]
      1. L,2
        1. <RCPPARNM1>
        2. <RCPPARVAL1>
        .
        .
      m. L,2
        1. <RCPPARNMm>
        2. <RCPPARVALm>
  5. <PRPROCESSSTARTp>
  6. <PRPAUSEEVENTp>

```

#### Exception

The list for specifying material (item 3a and 3b) is empty (L,0 instead of L,n), when no material is specified for the process job.  
The form of data item 3 (a or b) depends on the value in MF.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F16 PRJobMultiCreate Acknowledge	S,H<-E
<i>Description</i>	
This message acknowledges the request and reports any errors in the creation of a process job. ERRTEXT contains the identifier of process jobs that were not created.	
<i>Structure</i>	
<pre> L, 2   1. L, m                                [m = # jobs created]       1. &lt;PRJOBID<sub>1</sub>&gt;       .       .       m. &lt;PRJOBID<sub>m</sub>&gt;   2. L, 2       1. &lt;ACKA&gt;       2. L, n           1. L, 2               1. &lt;ERRCODE<sub>1</sub>&gt;               2. &lt;ERRTEXT<sub>1</sub>&gt;           .           .           n. L, 2               1. &lt;ERRCODE<sub>n</sub>&gt;               2. &lt;ERRTEXT<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F17 PRJobDequeue	S,H->E,reply
<i>Description</i>	
Used to remove process jobs from the equipment for jobs that have not begun processing.	
<i>Structure</i>	
<pre> L, m                                [m = # jobs to remove]   1. &lt;PRJOBID<sub>1</sub>&gt;   .   .   m. &lt;PRJOBID<sub>m</sub>&gt; </pre>	
<i>Exception</i>	
If m = 0, then de-queue all.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F18 PRJobDequeue Acknowledge	S,H<-E
<i>Description</i>	
Acknowledge the request to de-queue and report any errors. ERRTEXT will contain the identifier of any jobs that were not de-queued.	
<i>Structure</i>	
L, 2 1. L, m [m = # jobs removed] 1. <PRJOBID <sub>1</sub> > . . m. <PRJOBID <sub>m</sub> > 2. L, 2 1. <ACKA> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F19 PRGetAllJobs	S,H->E
<i>Description</i>	
Requests the equipment to return a list of process jobs which have not completed. They may be running or waiting to run.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F20 PRGetAllJobs Send	S,H<-E
<i>Description</i>	
Returns the requested list of process jobs.	
<i>Structure</i>	
L,m [m = # jobs in the list] 1. L,2 1. <PRJOBID <sub>1</sub> > 2. <PRSTATE <sub>1</sub> > . . m. L,2 1. <PRJOBID <sub>m</sub> > 2. <PRSTATE <sub>m</sub> >	
<i>Exception</i>	
If m = 0, then no process jobs are running or waiting to run.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F21 PRGetSpace	S,H->E
<i>Description</i>	
Requests the equipment to return the number of process jobs it has space to create.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F22 PRGetSpace Send	S,H<-E
<i>Description</i>	
Sends the host the number of process jobs which can be created.	
<i>Structure</i>	
<PRJOBSPACE>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F23 PRJobSetRecipeVariable	S,H->E
<i>Description</i>	
Reset the value of recipe variable parameters for a specific process job.	
<i>Structure</i>	
L, 2 1. <PRJOBID> 2. L, m [m = # recipe variables] 1. L, 2 1. <RCPPARNM <sub>1</sub> > 2. <RCPPARVAL <sub>1</sub> > . . m. L, 2 1. <RCPPARNM <sub>m</sub> > 2. <RCPPARVAL <sub>m</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F24 PRJobSetRecipeVariable Acknowledge	S,H<-E
<i>Description</i>	
Indicate the status of the request to set recipe variables. ERRTEXT will contain the RCPPARNM value for parameters that could not be reset.	
<i>Structure</i>	
L, 2 1. <ACKA> 2. L, n 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no errors exist.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F25 PRJobSetStartMethod	S,H->E
<i>Description</i>	
Used to request to change the start method (Manual start or Automatic start) for one or more process jobs.	
<i>Structure</i>	
L, 2 1. L, m [m = # of jobs; m >= 1] 1. <PRJOBID <sub>1</sub> > . . m. <PRJOBID <sub>m</sub> > 2. <PRPROCESSSTART>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F26 PRJobSetStartMethod Acknowledge	S,H<-E
<i>Description</i>	
Acknowledges request to set job start method and indicates any errors. ERRTEXT will contain the identifiers of any process jobs that did not accept the new process start method.	
<i>Structure</i>	
L, 2 1. L, k [k = # of jobs which are referenced in the request and that are now in the specified start method] 1. <PRJOBID <sub>1</sub> > . . k <PRJOBID <sub>k</sub> > L, 2 1. <ACKA> 2. L, n [n = # pf jobs for which the request failed] 1. L, 2 1. <ERRCODE <sub>1</sub> > 2. <ERRTEXT <sub>1</sub> > . . n. L, 2 1. <ERRCODE <sub>n</sub> > 2. <ERRTEXT <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F27 Control Job Command Request	S,H->E
<i>Description</i>	
Send a control job command to a control job.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;CTLJOBID&gt;</li> <li>2. &lt;CTLJOBCMD&gt;</li> <li>3. L, 2 <ol style="list-style-type: none"> <li>1. &lt;CPNAME&gt;</li> <li>2. &lt;CPVAL&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
3. L,2 IS L,0 for commands that do not need parameters.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F28 Control Job Command Acknowledge	S,H<-E
<i>Description</i>	
Indicates success or failure of command request to a control job. If applicable ERRTEXT shall contain information on specific command parameter names or values that caused the error.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;ACKA&gt;</li> <li>2. L, 2 <ol style="list-style-type: none"> <li>1. &lt;ERRCODE&gt;</li> <li>2. &lt;ERRTEXT&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
2. L,2 IS L,0 if no errors.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F29 PRSetMtrlOrder (PRJSMO)	S,H->E,reply
<i>Description</i>	
This message requests the equipment's Processing Management Service to use a specific strategy for the order in which materials are processed.	
<i>Structure</i>	
<PRMTRLORDER>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S16,F30 PRSetMtrlOrder Acknowledge (PRJSMOA)	S,H<-E
<i>Description</i>	
This message acknowledges the request for change to the material process strategy by reporting back the value requested, if correct.	
<i>Structure</i>	
<ACKA>	
<i>Exception</i>	
None	

10.21 *Stream 17 Equipment Control and Diagnostics* — This stream is a continuation of Stream 2.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F0 Abort Transaction (S17F0)	S,H<->E
<i>Description</i>	
Same form as S1F0.	
<i>Structure</i>	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F1 Data Report Create Request (DRC)	M,H->E,reply
<i>Description</i>	
Create a Data Report definition. This function allows the referencing of a Data Source for the items (variables or attributes) specified in the data report.	
<i>Structure</i>	
L, 4 1. <DATAID> 2. <RPTID> 3. <DATASRC> 4. L, n 1. <VID <sub>1</sub> > 2. <VID <sub>2</sub> > . . . n. <VID <sub>n</sub> >	
<i>Exception</i>	
DATAID is a zero length item when the request can be sent in a single block. If RPTID is a zero length item, then the equipment shall return a value in RPTID by which the host can then reference the report definition. If RPTID has a value, then the equipment shall retain this value by which the host can then reference the report definition.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F2 Data Report Create Acknowledge (DRCA)	S,H<-E
<i>Description</i>	
Equipment confirms creation of a Data Report and returns RPTID.	
<i>Structure</i>	
L, 2 1. <RPTID> 2. <ERRCODE>	
<i>Exception</i>	
If ERRCODE is a zero length item, then no error occurred.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F3 Data Report Delete Request (DRD)	S,H->E,reply
<i>Description</i>	
Delete one or more data reports. This shall cause those reports to be unlinked from any Event Reports to which they were linked. This shall cause the report to be excluded from any Trace Reports for in which it had originally been included.	
<i>Structure</i>	
L, n 1. <RPTID <sub>1</sub> > 2. <RPTID <sub>2</sub> > . . n. <RPTID <sub>n</sub> >	
<i>Exception</i>	
If this message is sent with a zero length list, then all reports shall be deleted.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F4 Data Report Delete Acknowledge (DRDA)	S,H<-E
<i>Description</i>	
Equipment confirms or indicates any errors on the request to delete Data Reports. All Data Reports which could be deleted shall be listed in the response and the associated error code shall be included in the list.	
<i>Structure</i>	
L, 2 1. <ACKA> 2. L, m 1. L, 3 1. <RPTID <sub>1</sub> > 2. <ERRCODE <sub>1</sub> > 3. <ERRTEXT <sub>1</sub> > . . m. L, 3 1. <RPTID <sub>m</sub> > 2. <ERRCODE <sub>m</sub> > 3. <ERRTEXT <sub>m</sub> >	
<i>Exception</i>	
If ACKA is TRUE, then no errors were encountered, meaning all report requests were completed successfully and a zero-length list (m = 0) shall be sent. If some reports could not be deleted, then their RPTIDs shall be given in a space separated list in ERRTEXT.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F5 Trace Create Request (TRC)	M,H->E,reply
<i>Description</i>	
Establish a Trace Report definition.	
<i>Structure</i>	
L, 6 1. <DATAID> 2. <TRID> 3. <CEED> 4. L, n 1. <RPTID <sub>1</sub> > 2. <RPTID <sub>2</sub> > . . n. <RPTID <sub>n</sub> > 5. <TRSPER> 6. L, m (m = {0,8}) 1. <TOTSMP> 2. <REPGSZ> 3. <EVNTSRC> (Defines source for start Event) 4. <CEID> (Defines ID of the start Event) 5. <EVNTSRC> (Defines source for stop Event) 6. <CEID> (Defines ID of the stop Event) 7. <TRAUTOD> 8. <RPTOC>	
<i>Exception</i>	
The list <i>m</i> can be zero-length, or it can contain all eight data items. Since specifying values for each item is optional, each of the eight items can be zero-length. If the item is zero-length, the format of the item shall be the same format used in other instances of the S17,F5 message where the value is not zero-length.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F6 Trace Create Acknowledge (TRCA)	S,H<-E
<i>Description</i>	
Equipment confirms creation of an Event Report and returns a TRID.	
<i>Structure</i>	
L, 2 1. <TRID> 2. <ERRCODE>	
<i>Exception</i>	
If ERRCODE is a zero length item, then no error occurred.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F7 Trace Delete Request (TRD)	S,H->E,reply
<i>Description</i>	
The host requests to delete one or more Trace Reports.	
<i>Structure</i>	
L, n 1. <TRID <sub>1</sub> > 2. <TRID <sub>2</sub> > . . n. <TRID <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F8 Trace Delete Acknowledge (TRDA)	S,H<-E
<i>Description</i>	
This message is required to inform the host when a Trace Report could not be deleted. This message does not need to be sent to confirm the successful deletion of a Trace Report. If the report is sent for a successfully deleted Trace Report, then the ERRCODE item length shall be set to zero.	
<i>Structure</i>	
L, 2 1. <ACKA> 2. L, m 1. L, 3 1. <TRID <sub>1</sub> > 2. <ERRCODE <sub>1</sub> > 3. <ERRTEXT <sub>1</sub> > . . m. L, 3 1. <TRID <sub>m</sub> > 2. <ERRCODE <sub>m</sub> > 3. <ERRTEXT <sub>m</sub> >	
<i>Exception</i>	
If ACKA is TRUE, then no errors were encountered, meaning all report requests were completed successfully and a zero-length list (m = 0) shall be sent. If some reports could not be deleted, then their TRIDs shall be provided in a space separated list in ERRTEXT.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F9 Collection Event Link Request (CELR)	M,H->E,reply
<i>Description</i>	
Establish a Collection Event Report definition with respect to a specific Event Source.	
<i>Structure</i>	
L, 4 1. <DATAID> 2. <EVNTSRC> 3. <CEID> 4. L, n (n is the number of Reports to be linked) 1. <RPTID <sub>1</sub> > 2. <RPTID <sub>2</sub> > . . n. <RPTID <sub>n</sub> >	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F10 Collection Event Link Acknowledge (CELA)	S,H<-E
<i>Description</i>	
Indicate the success or failure of a Collection Event Link Request.	
<i>Structure</i>	
L, 3 1. <EVNTSRC> 2. <CEID> 3. <ERRCODE>	
<i>Exception</i>	
Item 3 should be set to zero length to indicate success.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F11 Collection Event Unlink Request (CEUR)	S,H->E,reply
<i>Description</i>	
Request to unlink a specific Data Report from a Collection Event Report.	
<i>Structure</i>	
L, 3 1. <EVNTSRC> 2. <CEID> 3. <RPTID>	
<i>Exception</i>	
Item one can be zero length, in which case the default event source is assumed.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F12 Collection Event Unlink Acknowledge (CEUA)	S,H<-E
<i>Description</i>	
Indicates success or failure of a requested Unlink.	
<i>Structure</i>	
L, 4 <ol style="list-style-type: none"> <li>1. &lt;EVNTSRC&gt;</li> <li>2. &lt;CEID&gt;</li> <li>3. &lt;RPTID&gt;</li> <li>4. &lt;ERRCODE&gt;</li> </ol>	
<i>Exception</i>	
Item one can be zero length to indicate the default event source. Item 4 is set to zero length if the primary request was successful.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F13 Trace Reset Request (TRR)	S,H->E,reply
<i>Description</i>	
The Host requests the equipment to clear the data and reset the specified trace reports. If n = 0, then all defined Trace Objects will be reset.	
<i>Structure</i>	
L, n <ol style="list-style-type: none"> <li>1. &lt;TRID<sub>1</sub>&gt;</li> <li>2. &lt;TRID<sub>2</sub>&gt;</li> <li>.</li> <li>.</li> <li>n. &lt;TRID<sub>n</sub>&gt;</li> </ol>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S17,F14 Trace Report Reset Acknowledge (TRRA)	S,H<-E
<i>Description</i>	
This list in item 1 contains the identifiers of all the Trace Objects which were reset. If all Trace Objects are successfully reset, then ACKA shall be set to TRUE.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;ACKA&gt;</li> <li>2. L, m <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;TRID<sub>1</sub>&gt;</li> <li>2. &lt;ERRCODE<sub>1</sub>&gt;</li> <li>3. &lt;ERRTEXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>m. L, 3 <ol style="list-style-type: none"> <li>1. &lt;TRID<sub>m</sub>&gt;</li> <li>2. &lt;ERRCODE<sub>m</sub>&gt;</li> <li>3. &lt;ERRTEXT<sub>m</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	



<i>Exception</i>
If ACKA is TRUE, then no errors were encountered, meaning all report requests were completed successfully and a zero-length list (m = 0) shall be sent.
If some reports could not be reset, then their TRIDs shall be given in a space separated list in ERRTEXT.

10.22 *Stream 18 Subsystem Control and Data* — Messages exchanged between component subsystems and higher level controllers. Compared to similar messages exchanged between equipment and host, subsystem messages are less complex.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F0 Abort Transaction (S18F0)	S,H<->E
<i>Description</i>	
Used in lieu of an expected reply to abort a transaction. Function 0 is defined in every stream and has the same meaning in every stream.	
<i>Structure</i>	
Header only	
<i>Exception</i>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F1 Read Attribute Request (RAR)	S,H->E,reply
<i>Description</i>	
This message requests the current values of specified attributes of the subsystem component indicated in TARGETID.	
<i>Structure</i>	
<pre> L, 2   1. &lt;TARGETID&gt;   2. L, n                n = # attribute identifiers       1. &lt;ATTRID<sub>1</sub>&gt;       .       .       n. &lt;ATTRID<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
If n = 0, then all attributes of the target component are requested.	

<i>Stream, Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F2 Read Attribute Data (RAD)	S,H<-E
<i>Description</i>	
This message returns the current values of requested attributes and the current status of the requested component indicated in TARGETID. Attributes are returned in the order requested.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;TARGETID&gt;</li> <li>2. &lt;SSACK&gt;</li> <li>3. L, n <ol style="list-style-type: none"> <li>1. &lt;ATTRDATA<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> <li>n. &lt;ATTRDATA<sub>n</sub>&gt;</li> </ol> </li> </ol>	

<i>Exception</i>
Both n = 0 and s = 0 if the target component is unknown.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F3 Write Attribute Request (WAR)	S,H->E,reply
<i>Description</i>	
This message requests the subsystem to set the value of read/write attributes of the component specified in TARGETID.	
<i>Structure</i>	
L, 2 1. <TARGETID> 2. L, n 1. L, 2 1. <ATTRID <sub>1</sub> > 2. <ATTRDATA <sub>1</sub> > . . n. L, 2 1. <ATTRID <sub>n</sub> > 2. <ATTRDATA <sub>n</sub> >	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F4 Write Attribute Acknowledge (WAA)	S,H<-E
<i>Description</i>	
This message acknowledges the success of failure of the request to write attribute data to the subsystem indicated in TARGETID.	
<i>Structure</i>	
L, 3 1. <TARGETID> 2. <SSACK> 3. <STATUSLIST>	
<i>Exception</i>	
s = 0 if the target component is unknown.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F5 Read Request (RR)	S,H->E,reply
<i>Description</i>	
The host requests the subsystem indicated in TARGETID to read information. DATASEG may be used to indicate a specific section of data to be read. DATALENGTH is used to limit the amount of data for that section.	
<i>Structure</i>	
L, 3 1. <TARGETID> 2. <DATASEG> 3. <DATALENGTH>	
<i>Exception</i>	
If DATASEG and DATALENGTH are both omitted (are zero length items) then all data is requested. If DATALENGTH only is omitted, then all data within the indicated section is requested.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F6 Read Data (RD)	S,H<-E
<i>Description</i>	
This message is used to return requested information from the subsystem indicated in TARGETID or to acknowledge the results of the request.	
<i>Structure</i>	
L, 4 1. <TARGETID> 2. <SSACK> 3. <DATA> 4. L, s 1. <STATUS <sub>1</sub> > . . s. <STATUS <sub>s</sub> >	
<i>Exception</i>	
If TARGETID is unknown, then DATA is zero length.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F7 Write Data Request (WDR)	S,H->E,reply
<i>Description</i>	
This message requests to write data to the subsystem component indicated in TARGETID. DATASEG may be used to indicate a specific section of data to be written or overwritten.	
<i>Structure</i>	
L, 4 1. <TARGETID> 2. <DATASEG> 3. <DATALENGTH> 4. <DATA>	
<i>Exception</i>	
If DATASEG and DATALENGTH are both omitted (are zero length items) then all data is to be overwritten. If only DATALENGTH is omitted, then all data within the indicated section is to be written.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F8 Write Data Acknowledge (WDA)	S,H<-E
<i>Description</i>	
This message acknowledges the success or failure of writing data to the subsystem indicated in TARGETID.	
<i>Structure</i>	
L, 3 1. <TARGETID> 2. <SSACK> 3. <STATUSLIST>	
<i>Exception</i>	
s = 0 if and only if TARGETID is unknown.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F9 Read ID Request (RIR)	S,H->E,reply
<i>Description</i>	
This message is used to request the subsystem indicated by TARGETID to read an identifier.	
<i>Structure</i>	
<TARGETID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F10 Read ID Data (RID)	S,H<-E
<i>Description</i>	
This message returns a requested material identifier MID as read by the subsystem indicated in TARGETID.	
<i>Structure</i>	
L, 4	
1. <TARGETID>	
2. <SSACK>	
3. <MID>	
4. <STATUSLIST>	
<i>Exception</i>	
s = 0 if and only if TARGETID is unknown.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F11 Write ID Request (WIR)	S,H->E,reply
<i>Description</i>	
This message is used to request the subsystem indicated by TARGETID to write an identifier.	
<i>Structure</i>	
L, 2	
1. <TARGETID>	
2. <MID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F12 Write ID Acknowledge (WIA)	S,H<-E
<i>Description</i>	
This message acknowledges the success or failure of the subsystem specified in TARGETID in writing the ID.	
<i>Structure</i>	
L, 3	
1. <TARGETID>	
2. <SSACK>	
3. <STATUSLIST>	
<i>Exception</i>	
s = 0 if and only if TARGETID is unknown.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F13 Subsystem Command Request (SCR)	S,H->E,reply
<i>Description</i>	
This message is used to request the subsystem indicated in TARGETID to perform a specific action.	
<i>Structure</i>	
L, 3 1. <TARGETID> 2. <SSCMD> 3. L, n 1. <CPVAL <sub>1</sub> > . . n. <CPVAL <sub>n</sub> >	
<i>Exception</i>	
If n = 0, no parameters are provided.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F14 Subsystem Command Acknowledge (SCA)	S,H<-E
<i>Description</i>	
This message reports the results from the subsystem specified in TARGETID for the requested action.	
<i>Structure</i>	
L, 3 1. <TARGETID> 2. <SSACK> 3. <STATUSLIST>	
<i>Exception</i>	
s = 0 if and only if TARGETID is unknown.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F15 Read 2D Code Condition Request (R2DCCR)	S,H->E,reply
<i>Description</i>	
The host requests the subsystem indicated in TARGETID to read information on 2D Code Condition.	
<i>Structure</i>	
<TARGETID>	
<i>Exception</i>	
None	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S18,F16 Read 2D Code Condition Data (R2DCCD)	S,H<-E
<i>Description</i>	
This message is used to return requested information on 2D Code Condition from the subsystem indicated in TARGETID or to acknowledge the results of the request.	
<i>Structure</i>	
L, 5 1. <TARGETID> 2. <SSACK> 3. <MID> 4. <STATUSLIST> 5. <CONDITIONLIST>	
<i>Exception</i>	
If <SSACK> is failure, the length of the <CONDITIONLIST> must be zero.	

### 10.23 Stream 19 Recipe and Parameter Management

10.23.1 This section contains messages to support the management of recipes that include the following features:

- Self-documenting recipe component headers,
- Support for multi-part recipes (recipe components are called PDEs or Process Definition Elements),
- User-configured parameters, and
- Full assurance of byte-by-byte integrity of PDE content.

10.23.2 Some Recipe and Parameter Management terms are used in the explanations of Stream 19 messages. Here is a very brief definition of some terms used.

- *PDE* — Process Definition Element—a component of a recipe. A PDE includes an informational PDEheader and a PDEbody that is used for execution.
- *Recipe* — Instructions or data that direct equipment activities. A recipe is composed of one or more PDEs.
- *uid* — Unique identifier of a PDE.
- *gid* — Group id used to identify PDEs that are substitutable for one another. These are often different versions of a recipe component.
- *InputMap, OutputMap* — Data used to resolve references between PDEs in a multi-component recipe. These maps consist of a list of PDE references (group ids) with the corresponding uid that identifies the specific PDE that satisfies this reference.
- *Resolve* — Determination of all the components in a multi-part recipe. This is the process of creating an OutputMap that satisfies all the PDE references in a given recipe.
- *TransferContainer* — A group of PDEs or PDEheaders bound together as a single Stream 13 Data Set for transfer. The format for the TransferContainer is defined outside of this specification.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F0 Abort Transaction (S19F0)	S,H<->E
<i>Description</i>	
Used in lieu of an expected reply to abort a transaction. Function 0 is defined in every stream and has the same meaning in every stream.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F1 Get PDE Directory (GPD)	M,H<->E, reply
<i>Description</i>	
This message is used to request the list of PDEs maintained by the service provider. The client requests a subset of the full list by specifying filters (list L,m). The COMPARISONOPERATOR is used to compare the given attribute value with the current value of that attribute. Filters are AND'ed together. The client specifies the PDE attributes (list L,n) to be returned (in addition to identifier).	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. L,m (m = number of filters provided) <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;PDEATTRIBUTENAME<sub>1</sub>&gt;</li> <li>2. &lt;COMPARISONOPERATOR<sub>1</sub>&gt;</li> <li>3. &lt;PDEATTRIBUTEVALUE<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> </ol> </li> <li>m. L, 3 <ol style="list-style-type: none"> <li>1. &lt;PDEATTRIBUTENAME<sub>m</sub>&gt;</li> <li>2. &lt;COMPARISONOPERATOR<sub>m</sub>&gt;</li> <li>3. &lt;PDEATTRIBUTEVALUE<sub>m</sub>&gt;</li> </ol> </li> </ol> </li> <li>2. L,n (n = number of additional attributes to be returned per PDE) <ol style="list-style-type: none"> <li>1. &lt;PDEATTRIBUTE<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> <li>n. &lt;PDEATTRIBUTE<sub>n</sub>&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
If m = 0, all PDEs stored by the service provider are reported. If n = 0, no additional attributes are requested for each PDE (Note that UID is always returned in PDEDIRITEM).	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F2 PDE Directory Data (PDD)	M,H<->E
<i>Description</i>	
Returns the list of the PDEs, held by the service provider, that match the provided filter. The values of the specified attributes are included for each.	
<i>Structure</i>	
<pre> L, 3 1. &lt;DIRRSPSTAT&gt; 2. &lt;STATUSTXT&gt; 3. L,m (number of PDEs matching the request parameters)   1. L,2     1. &lt;UID<sub>1</sub>&gt;     2. L,n (n = number of PDE attributes returned)       1. L,2         1. &lt;PDEATTRIBUTE<sub>1</sub>&gt;         2. &lt;PDEATTRIBUTEVALUE<sub>1</sub>&gt;         .         .       n. L,2         1. &lt;PDEATTRIBUTE<sub>n</sub>&gt;         2. &lt;PDEATTRIBUTEVALUE<sub>n</sub>&gt;       .       .   m. L,2     1. &lt;UID<sub>m</sub>&gt;     2. L,n (n = number of PDE attributes returned)       1. L,2         1. &lt;PDEATTRIBUTE<sub>1</sub>&gt;         2. &lt;PDEATTRIBUTEVALUE<sub>1</sub>&gt;         .         .       n. L,2         1. &lt;PDEATTRIBUTE<sub>n</sub>&gt;         2. &lt;PDEATTRIBUTEVALUE<sub>n</sub>&gt;           </pre>	
<i>Exception</i>	
<p>m=0 indicates that no PDEs matched the filter or none exist on the service provider. n=0 if no additional attributes were requested in S19,F1.</p> <p>When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F3 Delete PDE (DPDE)	M,H->E, reply
<i>Description</i>	
Request that the specified PDEs be deleted from service provider's collection of PDEs.	
<i>Structure</i>	
<pre> L,n (where n &gt; 0)   1. &lt;UID<sub>1</sub>&gt;   .   .   n. &lt;UID<sub>n</sub>&gt;           </pre>	
<i>Exception</i>	
n = 0 is not allowed.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F4 Delete PDE Acknowledge (DPDEA)	M,H<-E
<i>Description</i>	
Acknowledge or error for each specified UID.	
<i>Structure</i>	
L,n (n = number of PDEs referenced in the S19,F3 message) <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>1</sub>&gt;</li> <li>2. &lt;DELRSPSTAT<sub>1</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>n</sub>&gt;</li> <li>2. &lt;DELRSPSTAT<sub>n</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>n</sub>&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
If S19,F3 contained a list of zero UIDs, this is an error. A zero-length list (n=0) is returned as acknowledgement of this error. When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F5 Get PDE Header (GPH)	M,H<->E, reply
<i>Description</i>	
Client requests that the headers of one or more PDEs be sent from the service provider.	
<i>Structure</i>	
L,n (where n > 0) <ol style="list-style-type: none"> <li>1. &lt;UID<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> <li>n. &lt;UID<sub>n</sub>&gt;</li> </ol>	
<i>Exception</i>	
n = 0 is not allowed.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F6 PDE Header Data (PHD)	M,H<->E
<i>Description</i>	
Response to the request for PDE header transfer. The list, L,n provides a status response for each of the requested PDEs. The PDEheaders of any PDEs with a GETRSPSTAT code of 0 will be returned in a TransferContainer (identified by the TCID) using Data Set Transfer (DST, Stream 13). The TCID is used as the DSNAME. Due to the SECS-II data item size restriction, the TransferContainer is not returned directly in this response.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;TCID&gt;</li> <li>2. L,n (n = number of PDEs referenced in the S19,F5 message) <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>1</sub>&gt;</li> <li>2. &lt;GETRSPSTAT<sub>1</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>n</sub>&gt;</li> <li>2. &lt;GETRSPSTAT<sub>n</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
<p>If S19,F5 contained a list of zero UIDs, then this is an error and a zero-length list (n=0) is returned as acknowledgement. In this case, no TransferContainer will be transferred.</p> <p>If no requested PDEs have a GETRSPSTAT code of 0, no TransferContainer will be transferred.</p> <p>If no TransferContainer is to be transferred (see the two exceptions above), then the TCID shall be a zero length item.</p> <p>When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.</p>	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F7 Get PDE (GPDE)	M,H<->E, reply
<i>Description</i>	
Client requests that one or more PDEs be sent from the service provider.	
<i>Structure</i>	
L,n (where n > 0) <ol style="list-style-type: none"> <li>1. &lt;UID<sub>1</sub>&gt;</li> <li>.</li> <li>.</li> <li>n. &lt;UID<sub>n</sub>&gt;</li> </ol>	
<i>Exception</i>	
n=0 is not allowed.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F8 PDE Data (PDED)	M,H<->E
<i>Description</i>	
Response to the request for PDE transfer. The list L,n provides a status response for each of the requested PDEs. Any PDEs with a GETRSPSTAT code of 0 will be returned in a TransferContainer (identified by the TCID) using Data Set Transfer (DST, Stream 13). The TCID is used as the DSNAME. Due to the data item size restriction, the TransferContainer is not returned directly in this response.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;TCID&gt;</li> <li>2. L,n (n = number of PDEs referenced in the S19,F7 message) <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>1</sub>&gt;</li> <li>2. &lt;GETRSPSTAT<sub>1</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>n</sub>&gt;</li> <li>2. &lt;GETRSPSTAT<sub>n</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
If S19,F7 contained a list of zero UIDs, then this is an error and a zero-length list (n=0) is returned as acknowledgement. In this case, no TransferContainer will be transferred. If no requested PDEs have a GETRSPSTAT code of 0, no TransferContainer will be transferred. If no TransferContainer is to be transferred (see the two exceptions above), then the TCID shall be a zero length item. When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F9 Request To Send PDE (RTSP)	S,H<->E, reply
<i>Description</i>	
Request for the service provider to determine whether transfer of a TransferContainer of a specified size can be initiated using the S19F11/F12 transaction. The TCID identifies the TransferContainer to be sent to the service provider.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;TCID&gt;</li> <li>2. &lt;TRANSFERSIZE&gt;</li> </ol>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F10 Send PDE Grant (SPDEG)	S,H<->E
<i>Description</i>	
Grants or denies the request to send PDEs. The TCID value shall match that in the corresponding S19,F9 message.	
<i>Structure</i>	
L, 3 <ol style="list-style-type: none"> <li>1. &lt;TCID&gt;</li> <li>2. &lt;RTSRSPSTAT&gt;</li> <li>3. &lt;STATUSTXT&gt;</li> </ol>	

<i>Exception</i>
When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F11 Send PDE (SPDE)	S,H<->E, reply
<i>Description</i>	
This message initiates the transport of a TransferContainer to the service provider. Restrictions on the size of a data item prevent the inclusion of the TransferContainer in this message. Instead, this message tells the service provider to initiate Data Set Transfer (DST, Stream 13) of the TransferContainer with DSNAME = TCID.	
<i>Structure</i>	
<TCID>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F12 Send PDE Acknowledge (SPDEA)	S,H<->E
<i>Description</i>	
Acknowledges receipt of the S19,F11 message. See S19,F13 for results of the transfer.	
<i>Structure</i>	
Header only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F13 TransferContainer Report (TR)	M,H<->E, reply
<i>Description</i>	
Acknowledges the receipt of a TransferContainer using Stream 13 Data Set Transfer. Verification of transferred PDE's is required upon receipt by equipment. Verification results are included in this message. The text in STATUSTXT reflects the status of transfer of the PDE referenced by the UID value. SENDRSPSTAT and VERIFYRSPSTAT together represent any error condition.	
<i>Structure</i>	
<pre> L, 3   1. TCID   2. SENDRESULT   3. L,n (where n &gt; 0; n = number of PDEs contained in the related TransferContainer)     1. L,4       1. &lt;UID<sub>1</sub>&gt;       2. &lt;SENDERSPSTAT<sub>1</sub>&gt;       3. &lt;VERIFYRSPSTAT<sub>1</sub>&gt;       4. &lt;STATUSTXT<sub>1</sub>&gt;     .     .   n. L,4     1. &lt;UID<sub>n</sub>&gt;     2. &lt;SENDERSPSTAT<sub>n</sub>&gt;     3. &lt;VERIFYRSPSTAT<sub>n</sub>&gt;     4. &lt;STATUSTXT<sub>n</sub>&gt; </pre>	

<i>Exception</i>
If the service provider is not an equipment, then verification will not occur and all VERIFYRSPSTAT will be zero length data items.
When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F14 PDE TransferContainer Report Acknowledge (TA)	S,H<->E
<i>Description</i>	
Acknowledges the receipt of the S19,F13 message.	
<i>Structure</i>	
Header Only	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F15 Resolve PDE Request (RPR)	M,H->E, reply
<i>Description</i>	
Request for the equipment to provide specific references (UIDs) for all PDEs in the (multi-part) recipe structure. The TARGETPDE is the apex of this structure. The second item in the initial list of the message structure "L,n" is the PDE InputMap. This provides client preferences for cases where indirect PDE references occur in the recipe structure.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;TARGETPDE&gt;</li> <li>2. L, n <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;PDEREF<sub>1</sub>&gt;</li> <li>2. &lt;RESOLUTION<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> </ol> </li> <li>n. L, 2 <ol style="list-style-type: none"> <li>1. &lt;PDEREF<sub>n</sub>&gt;</li> <li>2. &lt;RESOLUTION<sub>n</sub>&gt;</li> </ol> </li> </ol>	
<i>Exception</i>	
The PDE InputMap is included at the option of the client. n=0 if no InputMap is provided.	



<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19, F16 Resolve PDE Data (RPD)	M,H<-E
<i>Description</i>	
This message returns the OutputMap of the recipe structure (first list item “L,m” in the message structure) with the resolution of each PDereference found in the recipe structure. If a PDEREF is not resolved, it is not included in the OutputMap. The second list gives the response status for each PDEREF found in the recipe structure.	
<i>Structure</i>	
<pre> L, 2   1. L,m (m = number of PDE references in the recipe structure that were resolved)     1. L, 2       1. &lt;PDEREF<sub>1</sub>&gt;       2. &lt;RESOLUTION<sub>1</sub>&gt;       .       .     m. L, 2       1. &lt;PDEREF<sub>m</sub>&gt;       2. &lt;RESOLUTION<sub>m</sub>&gt;   2. L,n     1. L, 3       1. &lt;UID<sub>1</sub>&gt;       2. &lt;RESPDESTAT<sub>1</sub>&gt;       3. &lt;STATUSTXT<sub>1</sub>&gt;       .       .     n. L, 3       1. &lt;UID<sub>n</sub>&gt;       2. &lt;RESPDESTAT<sub>n</sub>&gt;       3. &lt;STATUSTXT<sub>n</sub>&gt; </pre>	
<i>Exception</i>	
n = the number of PDE reference errors found in the recipe structure or n = 1 if all PDE references were successfully resolved. That single entry represents the TARGETPDE with RESPDESTAT=0. When there is a positive result, StatusTxt is allowed to be a zero length data item.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F17 Verify PDE (VP)	M,H->E, reply
<i>Description</i>	
This message requests that a PDE be verified. If VERIFYDEPTH = 1 (All) then the second item in the message structure (L,n) represents the InputMap that contains user suggested resolutions for PDEREFs in the message structure.	
<i>Structure</i>	
L, 4 <ol style="list-style-type: none"> <li>1. &lt;TARGETPDE&gt;</li> <li>2. L, n <ol style="list-style-type: none"> <li>1. L, 2 <ol style="list-style-type: none"> <li>1. &lt;PDEREF<sub>1</sub>&gt;</li> <li>2. &lt;RESOLUTION<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 2 <ol style="list-style-type: none"> <li>1. &lt;PDEREF<sub>n</sub>&gt;</li> <li>2. &lt;RESOLUTION<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> <li>3. &lt;VERIFYTYPE&gt;</li> <li>4. &lt;VERIFYDEPTH&gt;</li> </ol>	
<i>Exception</i>	
The PDE InputMap is included at the option of the client. n=0 if no InputMap is provided. If VERIFYDEPTH = 0, then the list "L,n" (InputMap) is ignored and in this case, n should = 0.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F18 Verify PDE Data (VPD)	M,H<-E
<i>Description</i>	
Results of the requested verification.	
<i>Structure</i>	
L, 2 <ol style="list-style-type: none"> <li>1. &lt;VERIFYSUCCESS&gt;</li> <li>2. L, n (n = the number of PDEs for which verification was attempted) <ol style="list-style-type: none"> <li>1. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>1</sub>&gt;</li> <li>2. &lt;VERIFYRSPSTAT<sub>1</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>1</sub>&gt;</li> </ol> </li> <li>.</li> <li>.</li> <li>n. L, 3 <ol style="list-style-type: none"> <li>1. &lt;UID<sub>n</sub>&gt;</li> <li>2. &lt;VERIFYRSPSTAT<sub>n</sub>&gt;</li> <li>3. &lt;STATUSTXT<sub>n</sub>&gt;</li> </ol> </li> </ol> </li> </ol>	
<i>Exception</i>	
Note that if VERIFYDEPTH = 0, then n must = 1. When there is a positive result and no error condition is reported, StatusTxt is allowed to be a zero length data item.	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F19 RaP Multi-block Inquire (RMI)	S,H<->E, reply
<i>Description</i>	
Request permission to send one of the following multi-block messages: S19F1, S19F3, S19F5, S19F7, S19F13, S19F15, S19F17. This message is included for compatibility purposes.	
<i>Structure</i>	
<DATALENGTH>	
<i>Exception</i>	
None	

<i>Stream,Function Name (Mnemonic)</i>	<i>Direction</i>
S19,F20 RaP Multi-block Grant (RMG)	S,H<->E
<i>Description</i>	
Provides permission to send any of the corresponding multi-block messages (see S19F19 above).	
<i>Structure</i>	
<GRANT>	
<i>Exception</i>	
GRANT enumeration value of 3 is not valid for this message. S19F19 begins a conversation. That conversation is considered complete if any of the following messages are subsequently received: S19F1, S19F3, S19F5, S19F7, S19F13, S19F15, or S19F17. Note: the requestor is responsible for ensuring that the next Stream 19 message sent corresponds to the multi-block request.	

## 11 Message Documentation

11.1 *Intent* — Equipment makers using SECS-II messages must communicate the equipment-specific details of each message to the host designer in order for the host to properly adapt to the equipment. The details are communicated in a document which will follow a standard form in order to convey most clearly the information required. The following form is presented here to act as a guide for organizing the equipment-specific details.

11.2 *Standard Form SECS-II Document* — The standard form will contain three clearly labeled parts as follows.

- Part I — General Information
- Part II — Message Summary
- Part III — Message Detail

11.2.1 Part I will contain general information on the following:

- Manufacturer and product number
- General description of equipment function
- Intended function of interface
- Software revision code
- Changes from previous versions

11.2.2 Part II will contain two lists of all messages understood and all messages sent by the equipment in terms of their stream and function codes. The first list will have pairs of columns: the first for the message received and understood and the second for the message sent in response. The second list will also have two columns: the first for the message sent and the second for the response understood. The message will be identified using the format “SxxFyy,” where xx is the stream number and yy is the function number. Each transaction will be on a separate line. A “-” will indicate that one of a pair is not included. All messages not listed on the received side are implied to cause an error message to the host. All messages not listed on the sent side are assumed never to be sent from the equipment. Since some messages can be sent in two directions, the same message pair may appear in each list with the sent and received orders interchanged. A transaction listed in the standard as being allowed in two directions does not have to be implemented in both directions. This list will indicate which directions are implemented.

11.2.3 Part III will contain the details for every message listed in Part II. Messages that appear on both the sent and received sides must be detailed separately. The details shall include the following information on the data in each message:

1. For each fixed item, all the values or strings either understood or possible to send are listed along with their meaning to the equipment.
2. For each variable item, the restriction on or possible range of value or length of string.
3. Any other special interpretation of the message.

11.2.4 Each message so detailed must be clearly labeled with its stream and function code.

## 12 Units of Measure

12.1 *Intent* — Certain SECS-II transactions require specification of units of measure for data items passed between equipment and host. The concept of units of measure has been included as part of the SECS-II standard to enhance the ability of the host system to prompt its human operators for proper information when generating process programs, and also to facilitate automated handling of process programs by host systems and automated handling of data reported to a host by equipment.

12.2 *Units Symbols* — Under SECS-II, a units symbol is a text string of unspecified length which specifies the physical significance of a numeric value. Units of measure symbols under SECS-II may be either a SECS-II recognized unit identifier, a SECS-II unit identifier with prefix and/or suffix symbols, or an arithmetic expression of SECS-II identifiers.

12.2.1 A SECS-II units identifier is a text string which may be the full name, an abbreviation of the full name, or a special character which is unique for a specific unit of weight or measure. Identifier strings may consist of upper or lower case alphabetic characters and numerals or special characters of the ASCII character set. The first character of an identifier may not be a numeral. The case of alphabetic characters is significant (e.g., G and g, the units symbols for Gauss and gram, respectively).

12.2.2 A unit identifier may be nationally or internationally recognized, may be unique to the semiconductor industry, or, due to the special requirements of SECS-II, may be unique to this standard. § 12.4 lists all units identifiers recognized by SECS-II. For each identifier defined in § 12.4, six pieces of information are provided. They are:

1. *Unit* — Full name of the unit of measure in question.
2. *Unit Identifier* — SECS-II-recognized identifier for the unit.
3. *Prefix Allowed* — Specifies whether or not the unit identifier may be combined with a prefix symbol to generate a unit identifier which is a decimal multiple or submultiple of the base unit. Metric (or SI) units are usually capable of accepting a prefix symbol while English units may not.
4. *Suffix Allowed* — Specifies whether or not the unit identifier may be concatenated with a numeric suffix which provides additional information to the meaning of the associated unit symbol. The numeric suffix is composed of the ASCII digits 0 through 9 and represents a decimal value. This meaning of the numeric value is symbol-dependent and must be specified in the description section of the unit symbol's definition.
5. *Equivalence* — In those cases where a unit can be expressed as an arithmetic expression (of simpler units), this column will contain the expression of simpler units. For those units which are non-standard to either of the standard systems of units of measure (English or SI), this column will contain an expression which relates the non-standard unit to the equivalent unit of the standard units system. In either case, the expression provided in this column may be substituted for the corresponding SECS-II units identifier whenever required.
6. *Description* — Additional information as may be required to uniquely define the unit of measure in question.

12.2.3 Any SECS-II identifier which § 12.4 indicates as being capable of taking on a prefix symbol may be appended to one of the prefix symbols shown in Table 23, forming a new unit which is a decimal multiple or submultiple of the base unit. A prefix symbol may not be used alone. It must appear concatenated to one of the identifiers in § 12.4. Finally, only one prefix symbol may appear before any identifier. A units symbol such as “mus” (micromillisecond) is not allowed. The proper symbol is “ns” (nanosecond).

**Table 23**

<i>Prefix Name</i>	<i>Multiplicative Factor</i>	<i>Prefix Symbol</i>
exa	$10^{18}$	E
peta	$10^{15}$	P
Tera	$10^{12}$	T
giga	$10^9$	G
mega	$10^6$	M
kilo	$10^3$	k
hecto	$10^2$	h
deka (deca)	$10^1$	da
deci	$10^{-1}$	d
centi	$10^{-2}$	c
milli	$10^{-3}$	m
micro	$10^{-6}$	u
nano	$10^{-9}$	n
pico	$10^{-12}$	p
femto	$10^{-15}$	f
atto	$10^{-18}$	a

12.2.4 Any SECS-II identifier which § 12.4 indicates as being capable of taking on a suffix value may have a numeric string appended to it. This decimal value allows the user to identify one of a family of symbol names with only the generic symbol name of the family being defined in § 12.4. The meaning of a numeric suffix is dependent on the particular symbol with which it is being used and must be defined in the description section of the symbol definition.

12.2.5 Arithmetic expressions of units of measure identifiers are recognized by SECS-II as units symbols if they are formed by the following rules:

1. All units identifiers in the expression are SECS-II units identifiers defined in § 12.4 or SECS-II prefixed units identifiers as defined above.
2. Exponentiation is denoted by a circumflex (^) between the identifier to be operated on and the exponent. Exponents may be positive or negative values. A negative value is denoted by a unary minus sign (–) between the circumflex and exponent. For positive values, the exponent will immediately follow the circumflex ( $A^2$  or  $A^{-2}$ ).
3. Multiplication of units identifiers is expressed by an asterisk (\*) positioned between the factors to be multiplied ( $A*B$ ).
4. Division of units identifiers is expressed by a slash (/) positioned between the dividend and divisor. Division may also be expressed as the product of the dividend and the divisor with a negative exponent ( $A/B$  or  $A*B^{-1}$ ).
5. Parentheses may be used to specify the order in which the arithmetic operations will be performed.
6. Within expressions or sub-expressions where parentheses do not specify the order of operations, exponentiation will be carried out first, followed by left-to-right evaluation of all multiplication and division that is ( $A*B^{-2}*30*C^2$ ) is equivalent to  $((A/(B^2))*30)*(C^2)$ .

12.3 *Compliance* — For the units of measure information to have any value and to be in compliance with SECS-II, equipment and host system manufacturers must ensure that only units symbols allowed by SECS-II are used by their systems. In those instances where SECS-II does not provide a units symbol required for a particular application, the manufacturer requiring the new symbol may submit a proposal to the SEMI Communications Subcommittee requesting the enhancement. A proposal must include all the information provided by each entry of § 12.4 as described above.

12.3.1 A proposal must undergo the full approval cycle as prescribed by SECS-II for amending a standard (acceptance by committee, balloting, etc.). As a result, the proposal should be submitted as soon as possible, so that

sufficient time is available to complete the standard amendment process and to notify all interested parties of the change before the product requiring the new symbol becomes available for use in a manufacturing facility.

**12.4 SECS-II Units of Measure Identifiers** — All units of measure symbols recognized by SECS-II are defined in this section or are compound symbols based on the identifiers defined here and formed by the rules specified in § 12.2. Portions of the information provided below have been obtained from ANSI/IEEE 260-1978, ANSI X3.5-1976, ISO 2955-1974(E), Webster's New World Collegiate Dictionary (© 1977), and the CRC Handbook of Chemistry and Physics (52nd edition for 1971–1972).

**Table 24 SECS-II Units of Measure Identifiers**

<i>Unit</i>	<i>Identifier</i>	<i>Equivalence</i>	<i>Prefix Allowed</i>	<i>Suffix Allowed</i>	<i>Description</i>
Non-dimensional quantities (pure numbers)	null string	None	No	No	For all quantities which have no associated unit of measure, a zero length (null) text string is the appropriate 'identifier' to use when units of measure information is required.
ampere	A	None	Yes	No	SI unit of electric current.
ampere (turn)	AT	None	Yes	No	SI unit of magnetomotive force.
angstrom	Ang	m*10 <sup>-10</sup>	Yes	No	Unit of length used when measuring wavelength of light.
atmosphere, standard	atm	101325*Pa	No	No	A unit of pressure.
atmosphere, technical	at	kgf/cm <sup>2</sup>	No	No	A unit of pressure.
atomic mass unit (unified)	u	1.660531*10 <sup>-27</sup> *kg	No	No	One twelfth the mass of an atom of carbon 12 nuclide.
bar	bar	100*kPa	Yes	No	CGS unit of pressure.
barn	barn	10 <sup>-28</sup> *m <sup>2</sup>	Yes	No	Unit for measuring capture cross sections of elements.
barrel (petroleum)	bbl	42*gal or 158.99*l	No	No	A unit of volume.
baud	Bd	bit/s	Yes	No	Telecommunications measure of data transfer rate equivalent to one bit of information transferred per second.
bel	B	None	Yes	No	The logarithm of the ratio of two power signals.
Becquerel	Bq	None	Yes	No	SI unit of activity of a radionuclide.
bit	bit	None	Yes	No	A unit of computer information equivalent to the choice between two alternatives (as yes or no, on or off).
boat	boat	None	No	Yes	Special SECS generic unit corresponding to a holder of wafers or packages with discrete positions. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
British thermal unit	Btu	1054.35*J	No	No	The quantity of heat required to raise the temperature of one pound of water one degree Fahrenheit at or near 39.2°F.
byte	byte	8*bit	Yes	No	Unit of storage for computer memory.
calorie (International Table)	callIT	4.1868*J	Yes	No	Defined by the 1929 International Stream Table Conference to be 1/860 international joules or 1/859.858 joules.
calorie (thermochemical)	cal	4.1840*J	Yes	No	Unit of energy defined by the NBS to be 4.184 joules. Also, called the gram calorie.
candela	cd	None	Yes	No	SI unit of luminous intensity.
candle	cd	None	Yes	No	Alternate name for candela.

<i>Unit</i>	<i>Identifier</i>	<i>Equivalence</i>	<i>Prefix Allowed</i>	<i>Suffix Allowed</i>	<i>Description</i>
carrier	carrier	None	No	Yes	Special SECS generic unit corresponding to a holder for substrates, wafers or wafer frames. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
cassette	css	None	No	Yes	Special SECS generic unit corresponding to a holder for wafers or wafer frames. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
Coulomb	C	A*s	Yes	No	SI unit of electric charge.
curie	Ci	$3.7 \times 10^{10} \text{ Bq}$	No	No	A unit of activity of radionuclide.
cycle	c	None	Yes	No	Unit equivalent to one complete performance of a periodic process.
darcy	D	$\text{cP} \cdot (\text{cm/s}) / (\text{cm/atm})$ or $0.986923 \cdot \mu\text{m}^2$	No	No	Unit of permeability of a porous medium. By traditional definition, a permeability of one darcy will permit a flow of $1 \text{ cm}^3/\text{s}$ of fluid of $1 \text{ cP}$ viscosity through an area of $1 \text{ cm}^2$ under a pressure gradient of $1 \text{ atm/cm}$ .
day (mean solar)	d	24*h	No	No	The period required for the Earth to complete one rotation about its axis.
degree (plane angle)	deg	$\pi/180 \cdot \text{rad}$	No	No	One three hundred sixtieth part of the circumference of a circle.
degree Celsius	degC	None	No	No	Unit of temperature where $0^\circ\text{C}$ corresponds to the freezing point of water and $100^\circ\text{C}$ corresponds to the boiling point at standard atmospheric conditions.
degree Fahrenheit	degF	None	No	No	Unit of temperature where $32^\circ\text{F}$ corresponds to the freezing point of water and $212^\circ\text{F}$ corresponds to the boiling point at standard atmospheric conditions.
degree Kelvin	K	None	No	No	SI unit of temperature.
die	die	None	No	No	Special SECS generic unit corresponding to an individual integrated circuit both on a wafer and after wafer separation. Also referred to as a bar or chip.
dyne	dyn	$10^{-5} \cdot \text{N}$	Yes	No	Unit of force in the cgs system. One dyne is the force required to provide a one grain mass with an acceleration of $1 \text{ cm/s}^2$ .
electronvolt	eV	$1.60209 \times 10^{-19} \cdot \text{J}$	Yes	No	Energy acquired by a small particle carrying a unit electronic charge when it falls through a potential difference of one volt.
erg	erg	$10^{-7} \cdot \text{J}$	Yes	No	Unit of work or energy in the cgs system. One erg is equal to the work done or energy expended to exert a force of one dyne through a distance of 1 cm.
farad	F	A*s/V	Yes	No	SI unit of capacitance.
foot	ft	12*in	No	No	English unit of length.
footcandle	Fc	$\text{lm/ft}^2$	No	No	Unit of illuminance. Also called lumen per square foot.

<i>Unit</i>	<i>Identifier</i>	<i>Equivalence</i>	<i>Prefix Allowed</i>	<i>Suffix Allowed</i>	<i>Description</i>
footlambert	FL	$(1/\pi) \cdot \text{cd}/\text{ft}^2$	No	No	A unit of luminance. One lumen per square foot leaves a surface whose luminance is one footlambert in all directions within a hemisphere.
gal	Gal	$\text{cm}/\text{s}^2$	Yes	No	A unit of acceleration used especially for values of gravity.
gallon (US)	gal	$231 \cdot \text{in}^3$ or $4 \cdot \text{qt}$ or $3.7854 \cdot \text{l}$	No	No	United States version of English system unit of volume.
gallon (UK)	galUK	$4.5461 \cdot \text{l}$	No	No	United Kingdom version of English system unit of volume.
gauss	G	$\text{Mx}/\text{cm}^2$	Yes	No	Electromagnetic CGS unit of magnetic flux density.
gilbert	Gb	$10/(4\pi) \cdot \text{AT}$	Yes	No	Electromagnetic CGS unit of magnetomotive force.
grain	gr	$0.0022857143 \cdot \text{oz}$	No	No	English unit of weight.
gram	g	None	Yes	No	One thousandth of the SI unit of mass.
gram-force	gf	$9.80665 \cdot \text{N} \cdot 10^{-3}$	Yes	No	The weight of a gram mass when subjected to the mean gravitational attraction of the Earth.
gray	Gy	Unknown	Yes	No	SI unit of absorbed dose in the field of radiation dosimetry.
henry	H	$\text{V} \cdot \text{s}/\text{A}$	Yes	No	SI unit of inductance.
hertz	Hz	$\text{c}/\text{s}$	Yes	No	SI unit of frequency.
horsepower (electric)	hp	$746 \cdot \text{W}$	No	No	Archaic unit of power.
hour	h	$60 \cdot \text{min}$	No	No	Derived unit of time.
inch	in	$2.54 \cdot \text{cm}$	No	No	English unit of length.
conventional inch of mercury	inHg	$3386.4 \cdot \text{Pa}$	No	No	Unit equivalent to the pressure required to balance a one inch high column of mercury in a manometer at 32°F.
conventional inch of water	inH <sub>2</sub> O	$249.09 \cdot \text{Pa}$	No	No	Unit equivalent to the pressure required to balance a one inch high column of water in a manometer at 4°C.
ingot	ing	None	No	No	Special SECS generic unit corresponding to the entity of semiconductor manufacture from which wafers are made.
ion	ion	None	No	No	SECS II unique symbol equivalent to an atom that carries an electric charge as a result of losing or gaining electrons.
joule	J	$\text{N} \cdot \text{m}$	Yes	No	SI unit of energy, work, and quantity of heat.
kelvin	K	None	No	No	SI unit of temperature. Also referred to as degree Kelvin.
kilopound force	klbf	$1000 \cdot \text{lbf}$	No	No	A multiple of the English unit of force or weight.
knot	kn	$\text{nmi}/\text{h}$	No	No	Unit of velocity expressed in nautical miles per hour.
lambert	L	$(1/\pi) \cdot \text{cd}/\text{cm}^2$	Yes	No	CGS unit of luminance. One lumen per square centimeter leaves a surface whose luminance is one lambert in all directions within a hemisphere.



<i>Unit</i>	<i>Identifier</i>	<i>Equivalence</i>	<i>Prefix Allowed</i>	<i>Suffix Allowed</i>	<i>Description</i>
leadframe	ldfr	None	No	Yes	Special SECS generic unit corresponding to a structure for leads which is removed after packaging. The structure may be fixed length or a reel. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
liter	l	$10^{-3} \text{m}^3$	Yes	No	A metric unit of volume.
lot	lot	None	No	No	Special SECS generic unit corresponding to a grouping of material which is undergoing the same processing operations. The amount of material represented by "1 lot" is situation-dependent.
lumen	lm	$\text{cd} \cdot \text{sr}$	Yes	No	SI unit of luminous flux.
lux	lx	$\text{lm}/\text{m}^2$	Yes	No	SI unit of illuminance.
magazine	mgz	None	No	Yes	Special, SECS generic unit corresponding to a holder of fixed length leadframes. The unit capacity is specified by the symbols suffix, if provided. Otherwise, the capacity is situation-dependent.
maxwell	Mx	$10^{-8} \text{Wb}$	Yes	No	Electromagnetic CGS unit of magnetic flux.
meter	m	None	Yes	No	SI unit of length.
metric ton	t	$10^3 \text{kgf}$	No	No	Unit of weight of force.
mho	mho	S	Yes	No	Previous name for the SI unit siemens.
micron	um	$10^{-6} \text{m}$	No	No	Alternate name for a micrometer.
conventional micron of mercury	umHg	$133.32 \text{Pa} \cdot 10^{-3}$	No	No	Unit of pressure.
mil	mil	$10^{-3} \text{in}$	No	No	English unit of length.
mile	mi	$5280 \text{ft}$	No	No	English unit of length.
conventional millimeter of mercury	mmHg	$133.322 \text{Pa}$	No	No	Unit of pressure.
millimicron	nm	$10^{-9} \text{m}$	No	No	Alternate name for nanometer.
minute (plane angle)	mins	$\text{deg}/60$	No	No	One sixtieth of a degree (plane angle).
minute(time)	min	$60 \cdot \text{s}$	No	No	Unit of time.
mole	mol	$6.02252 \cdot 10^{23}$	No	No	SI unit of number of entities within a substance.
month	mo	None	No	No	Unit of time.
nautical mile	nmi	$1852 \text{m}$	No	No	English unit of measurement.
neper	Np	$0.1151 \cdot \text{dB}$	Yes	No	Unit for expressing ratios of power levels.
newton	N	$\text{kg} \cdot \text{m}/\text{s}^2$	Yes	No	SI unit of force.
nit	nt	$\text{cd}/\text{m}^2$	Yes	No	Alternate name for the SI unit of luminance, candela per square meter.
oersted	Oe	$79.577472 \cdot \text{A}/\text{m}$	Yes	No	Electromagnetic CGS unit of magnetic field strength.
ohm	ohm	V/A	Yes	No	SI unit of resistance.
ounce (avoirdupois)	oz	$\text{lbf}/16$	No	No	English unit of weight.
package	pkg	None	No	No	Special SECS generic unit corresponding to an individual entity both as a place for the die to reside and as a completed unit.
pascal	Pa	$\text{N}/\text{m}^2$	Yes	No	SI unit of pressure or stress.

<i>Unit</i>	<i>Identifier</i>	<i>Equivalence</i>	<i>Prefix Allowed</i>	<i>Suffix Allowed</i>	<i>Description</i>
percent	%	1/100	No	No	Ratio of parts per hundred.
phot	ph	lm/cm <sup>2</sup>	Yes	No	CGS unit of illuminance.
pH	pH	1	No	No	Normalized measure of acidity or alkalinity.
pint (UK)	ptUK	0.56826*1	No	No	United Kingdom version of English unit of capacity.
pint (US dry)	ptUS	0.55061*1	No	No	United States version of English unit of dry capacity.
pint (US liquid)	pt	0.47318*1	No	No	United States version of English unit of liquid capacity.
plate	plt	None	No	Yes	Special SECS generic unit corresponding to a temporary fixture used to hold die during assembly operations. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
poise	P	36*N*s/m <sup>2</sup> , or 36*kg/(m*s)	Yes	No	A CGS unit of viscosity equal to the viscosity of a fluid that would require a shearing force of one dyne to move a square centimeter area of either of two parallel layers of fluid one centimeter apart with a velocity of one centimeter per second relative to the other layer, with the space between the layers being filled with fluid.
pound	lb	0.0310810*slug	No	No	English unit of mass.
pound-force	lbf	4.4482217*N	No	No	English unit of force or weight.
poundal	pdl	0.0310810*lbf	No	No	Force required to accelerate a one pound mass at one ft/s <sup>2</sup> .
parts per million	ppm	1/10 <sup>6</sup>	No	No	Ratio of parts per million.
quart (UK)	qtUK	1.1365*1	No	No	United Kingdom version of an English unit of capacity.
quart (US dry)	qtUS	1.1012*1	No	No	United States version of an English unit of dry capacity.
quart (US liquid)	qt	0.94635*1	No	No	United States version of an English unit of liquid capacity.
rad	rd	10 <sup>-2</sup> *Gy	Yes	No	A unit of absorbed dose in the field of radiation dosimetry.
radian	rad	None	Yes	No	SI unit of plane angle.
rem	rem	10 <sup>-2</sup> *Sv	Yes	No	A unit of dose equivalent in the field of radiation dosimetry.
revolution	r	c	No	No	One complete cycle of a rotating body.
roentgen	R	Unknown	No	No	A unit of exposure in the field of radiation dosimetry.
second (plane angle)	sec	mins/60	No	No	One sixtieth of a minute of a degree.
second (time)	s	None	Yes	No	SI unit of time.
siemens	S	1/ohm	SI unit of conductance.	Yes	No.
sievert	Sv	Unknown	Yes	No	SI unit of dose equivalent in the field of radiation dosimetry.
slug	slug	14.5939*kg	No	No	English unit of mass.

<i>Unit</i>	<i>Identifier</i>	<i>Equivalence</i>	<i>Prefix Allowed</i>	<i>Suffix Allowed</i>	<i>Description</i>
standard cubic centimeter per minute	sccm	cc/min	No	No	A unit of flow equivalent to one cubic centimeter of a gas at standard temperature and pressure flowing past a point in one minute.
standard liter per minute	slpm	l/min	No	No	A unit of flow equivalent to one liter of a gas at standard temperature and pressure flowing past a point in one minute.
steradian	Sr	Unknown	Yes	No	SI unit of solid angle.
stilb	sb	cd/cm <sup>2</sup>	Yes	No	A CGS unit of luminance.
stokes	St	P*cm <sup>3</sup> /g	Yes	No	A CGS unit of kinematic viscosity.
substrate	substrate	None	No	No	Special SECS generic unit corresponding to the entity of material being operated on, processed or fabricated.
tesla	T	N/(A*m) or Wb/m <sup>2</sup>	Yes	No	SI unit of magnetic flux density (magnetic induction).
therm	thm	10 <sup>5</sup> *Btu	No	No	An English unit of energy.
ton (short)	ton	2000*lbf	No	No	English unit of weight.
torr	torr	mmHg	Yes	No	Pressure unit. Alternative name for millimeters of mercury.
tube	tube	None	No	Yes	Special SECS generic unit corresponding to a holder of packages arranged in a flow. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
var	var	Unknown	Yes	No	SI unit for reactive power.
volt	V	W/A	Yes	No	SI unit of voltage.
wafer	wfr	None	No	No	Special SECS generic unit corresponding to the entity of material on which semiconductor devices are fabricated.
waferframe	wffr	None	No	Yes	Special SECS generic unit corresponding to a temporary fixture for wafers. The unit capacity is specified by the symbol's suffix, if provided. Otherwise, the capacity is situation-dependent.
watt	W	J/s	Yes	No	SI unit of power.
watthour	Wh	3600*J	Yes	No	Unit of energy.
weber	Wb	V*s	Yes	No	SI unit of magnetic flux.
year	yr	None	No	No	Unit of time.

## RELATED INFORMATION 1

### APPLICATION NOTES

**NOTICE:** This related information is not an official part of SEMI E5 and was derived from the global Information & Control Committee. This related information was approved for publication by full letter ballot procedures.

#### **R1-1 The General Node Transaction Protocol**

R1-1.1 This application note has been moved to follow SEMI E4 (SECS-I) as § R1-7.

#### **R1-2 Some Suggested Message Usage**

R1-2.1 The number of messages implemented and the choice of messages are greatly influenced by the actual function of the equipment. To illustrate which messages might be appropriate, the following suggestions are offered for a variety of different types of equipment capabilities. It is assumed that the minimum message sets S1,F1; F2 and S9,F1; F3; F5; F7 are always implemented.

R1-2.2 For equipment which makes nondestructive in-process measurements using a fixed measurement procedure, it may be necessary only to implement S6,F9 to send the data according to a fixed format upon measurement. Optional remote control can be added with S2,F21 to start a measurement.

R1-2.3 If the equipment has a variety of measurement routines, it might be desirable to respond to S1,F5 with S1,F6, which would give the host a brief report of the test being made. The test can be thought of as a process program. Accordingly, S7,F1 and S7,F2 could be used for the host to select the program. The same messages in conjunction with S7,F3 and S7,F4 could load a new test procedure. S7,F19 could be used by the host to find out what tests were available.

R1-2.4 Some equipment which automatically processes wafers in a batch might make more extensive use of S1,F5 or S1,F3 and might include some error reporting on S5,F1. More sophisticated equipment may include some trace features with S2,F23 and S6,F1 or some control loop tuning by S2,F15.

R1-2.5 Equipment using in-line wafer movement could utilize Stream 4, S1,F9, and Stream 3 to keep track of wafers.

R1-2.6 Stream 7,F9 through F19 can be used to manage a local backup of process programs should the host fail for a short while.

R1-2.7 Microprocessor equipment can benefit from features such as provided by Stream 8 and S2,F1 through F12 which allow managing and servicing the software routines.

R1-2.8 Equipment, including a CRT, might elect to make it available to the host by including Stream 10 messages.

R1-2.9 Some equipment, such as functional testers, might have sufficient need to undertake remote file usage such as provided in Stream 13.

R1-2.10 These brief suggestions serve to illustrate that the final choice of the messages included in a given equipment depends upon its function. The messages can be viewed as interface features in the same way that other parts of the equipment are viewed as processing features or wafer handling features.

#### **R1-3 Notes on SECS-II Data Transfers**

##### *R1-3.1 Introduction*

R1-3.1.1 There are two primary ways to send and ask for data in SECS-II. One of these is to use the trace feature and the other is to use the event reporting method. The purpose of this note is to describe the intended operation of the messages described in the existing standard. Discussion of completeness or need for other reporting methods is left for task force and committee work.

##### *R1-3.2 Trace Data Collection and Reporting*

R1-3.2.1 This method of collecting data is intended for engineering and developmental use rather than routine data collection for production. The features included allow the collection of relatively large amounts of real time data over a finite amount of time. The data is generated at regular time intervals as determined by a timing generator in

the equipment. The function of the host is to set up the trace and then to subsequently store the data as it is received from the equipment. It is assumed that some host resident applications will exist to analyze the data either as it is received or at some later time.

R1-3.2.2 The trace feature will only exist in equipment which implements it.

R1-3.2.3 The host sets up the trace with the S2,F23–24 transaction. At this time, the host assigns several important parameters. TRID is the trace request ID and is used later when the equipment sends back the data. Every trace data reply includes the TRID corresponding to the request that set up the trace. Several traces can theoretically be done at the same time if the equipment allows it and the TRID keeps the data for each trace distinct from other trace data. DSPER is the data sample period and is used to indicate how often the specified parameters should be sampled (that is, have their values saved). TOTSMP is the total number of samples to be made. Since TOTSMP is finite and the number of parameters is specified in this transaction, the host can reserve adequate file space for the reported data if required. The REPGSZ is the reporting group size and corresponds to the number of time samples that should be combined into one message prior to transmission. Thus, if it is desired to sample one or two parameters every second but only send those samples to the host once a minute, the reporting group size would be 60. Having the reporting group size parameter allows the host to have some control over how often it may be interrupted to handle the trace data. However, as presently defined in the standard, the trace data is reported as a single block message (S6,F1), which restricts both the number of status variables or the number of samples which can be combined into one message. The equipment may be able to accommodate this in several ways, to be described shortly.

R1-3.2.3.1 The last element in the trace initialize request is a list of status variable IDs. The trace command only allows tracing variables that have been declared and are known to the equipment as status variables. It is assumed that the equipment will report the variable values in the same order as specified in the trace request. This will allow the host to identify the values returned.

R1-3.2.4 The trace data send message, S6,F1–2, sends the trace data as a single block message to minimize the overhead in reporting data. The TRID is the first item and identifies the request that asked for the data. The next item is SMPLN, the sample number of the last sample in this message, should more than one sample be combined. The next item is STIME which is the time of the last sample in this message. These three items are followed by the list of values. If five (5) values were requested with a reporting group size of 5, then 25 values would be in this list, each group of 5 in the same order as requested and in the time order sampled. Some flexibility is allowed in how the equipment chooses to report the data to the host when the reporting group size exceeds one block of data. The equipment can send the data when it has a complete block or it can reject the request when it is set up.

### R1-3.3 *Event Driven Data Reporting*

R1-3.3.1 The second major type of data reporting is initiated by some event in the equipment. Data reporting is often desired after some event such as the completion of a measurement, the completion of a lot, the completion of a wafer, the occurrence of a special event command in the recipe, or some other action which is determined by the equipment. The two aspects involved in event driven data reporting are the control of which events cause data to be sent to the host and the formatting of the data sent to the host.

R1-3.3.2 It is assumed that a set of events has been established for a particular piece of equipment and that each event can produce a report of some sort. It is further assumed that a set of equipment constants exist in the machine such that they have control over the optional reporting of the events. For example, a Boolean constant may exist for each possible reporting event, and when the host sets the constant to a logical 1, the corresponding event will cause a report to be sent to the host, when the host sets the constant to logical 0, the event will not send a report. S2,F15–16 in the equipment constant send transaction can be used to control the event reporting.

R1-3.3.3 When an event causes data to be sent to the host, there are several possible conversations, depending upon the length of the data and the complexity of the formatting. S6,F3–4 is the basic data transaction, which has a very general format. The parameters provide for an overall name, DATAID, for the type of data; a collection event identification, CEID, should there be more than one event that could generate the same type of data; and a list of data sets. This structure allows reporting such data as the measurements taken on each of the wafers in a lot. The measurements on each wafer make up one data set, and the list of data sets is the whole lot. The collection event would be the lot completion, and the data ID might be film thickness measurements. Other types of organizations are possible, depending upon the type of data being sent. The same type of data might be produced by a different CEID, such as the forced termination of the lot. This collection ID would indicate that the data is incomplete for the

lot. Within each data set, each data value is reported as a pair of items, one item being the name of the value and the other being the value.

R1-3.3.4 Since many simple measuring devices have only a very few types of data sets, an alternative data format is provided in S6,F9–10, which has the same form as S6,F3–4 but does not require the value name in the data set. Instead, the order of the values is fixed format for the particular DSID in that particular equipment.

R1-3.3.5 When either of the above data messages is long enough to require multiple blocks, it must be preceded by S6,F5–6 to gain permission to send a multiple block message.

R1-3.3.6 The last data control transaction is S6,F7–8, which is initiated by the host and causes a specified DATAID to be sent to the host. The implementation of this function is highly equipment-dependent. In essence, it is equivalent to the host causing an event that triggers the sending of the data. Since the equipment may be generating the data, the actual data sent depend upon the equipment implementation. The equipment can respond with a zero length item if no data can be sent.

#### R1-3.4 *Event Reporting*

R1-3.4.1 The third major type of data reporting is similar to that described in § R1-3.3 above, with the following enhancements:

1. Contents of data reports are not limited to DVVALs, but may include SVs or even ECVs.
2. Contents of data reports are user programmable.

R1-3.4.2 It is assumed that the equipment vendor supplies a list of all “events” identified within a particular piece of equipment. A Collection Event Identifier (CEID) must be specified for each of these events. It is further assumed that the vendor supplies a list of all available variables within the machine. This includes Status Variables (SVs) and their identifiers (SVIDs), Equipment constants (ECVs) and their identifiers (ECIDs), and Data Values (DVVALs) and their identifiers (DVNAMEs). Each of the identifiers must be unique. The term VID (Variable Identifier) encompasses all SVIDs, ECIDs, and DVNAMEs. Likewise, the term V (Variable Data) encompasses all SVs, ECVs, and DVVALs.

R1-3.4.3 Note that a Variable (V) may be a list (format code 0). This provides for referencing a group of related data values with one identifier. Consider the following:

```

VID1 = 1 zone 1 temperature ID   Format 32
V1   zone 1 temperature value Format 52
.
.
VIDn = n zone n temperature ID   Format 32
Vn   zone n temperature value Format 52
VIDx = x all temperatures ID     Format 32
Vx  L,n                         Format 0
      1. L,2
          1. <VID1>
          2. <V1>
      .
      .
      n. L,2
          1. <VIDn>
          2. <Vn>

```

Any V in a list may also be a list (for nesting).

R1-3.4.4 In a typical initialization sequence, the host would define all the desired programmable data reports with S2,F33/S2,F34 (Define Report) transactions. Then S2,F35/S2,F36 (Link Report/Event) transactions would be used to define which reports are to be made by the equipment upon specific events (CEIDs). An individual report may be linked to more than one event. At this point the host may request reports with the S6,F15/S6,F16 (Report Request) transactions to obtain initial report data and/or to verify reports as defined and linked. Finally, the desired reports would be enabled by the host with S2,F37/S2,F38 (Enable/Disable Event Report) transactions.

R1-3.4.5 There are two methods for the equipment to send event reports to the host. S6,F13 includes the Variable Identifier (VID) with each Variable Data item (V). S6,F11 is a shorter form, without the identifiers; some users prefer this form to reduce message size.

R1-3.4.6 When any message is long enough to require multiple blocks, it must be preceded by an inquire/grant transaction. The DATAID parameter is used only to link the inquire/grant transaction with a multiblock message. This linkage is to alleviate problems in the case of interleaved messages. A unique value for DATAID must be used for each Inquire/Grant/Send/Acknowledge conversation (similar to the use of SYSBYTES in SECS-I). The DATAID parameter should not be used for any other purpose.

## **R1-4 Process Programs**

### *R1-4.1 Introduction*

R1-4.1.1 Two forms of process programs are supported by SECS-II: unformatted and formatted. The contents of an unformatted process program conform to no set standard. The format of the program is defined by the vendor of the equipment and probably bears no similarity to the format used by other vendors for their equipment. Because special programming would be required at the host to understand the equipment's unique data format, the process program is most likely generated at the machine and the host is only used as a data repository, saving the foreign data for later retransmission to the equipment. S7,F3 and S7,F6 are the SECS-II messages used to move unformatted process programs between host and equipment.

R1-4.1.2 Unformatted process programs were the original accepted means for moving processing instructions between host and equipment under SECS-II. However, the inability of a host to generate process programs for its subordinate machines was quickly recognized as a severe problem. As a result, the formatted process program and its associated transactions were added. Five transactions are provided under SECS-II for handling formatted process programs: S7,F23–24, S7,F25–26 allow movement of process programs between host and equipment; S7,F21–22 originates at a machine and provides a host with the information it needs to generate a process program for that machine; S7,F27–28 allows the equipment to tell the host whether or not the contents of the formatted process program received from the host are valid; and S7,F31–32 provides the host with the ability to ask the equipment to check the validity of a process program without actually downloading the program into the machine for production use.

### *R1-4.2 Normal Sequence of Operations*

R1-4.2.1 Formatted process programs may be generated at a host or machine. The actions taken to generate one in a machine are left to the equipment manufacturer. If the process program is created at a host, a sequence of operations is assumed.

R1-4.2.1.1 Once the host's process program generator has been invoked and has been told for which machine a process program is to be created, the host editor must obtain a copy of the process capabilities data for that machine. The information may already be available on the host or it may be obtained directly from the machine. In either case, the information originates at the equipment and is obtained using S7,F21–22. (See § R1-4.4 for additional information.)

R1-4.2.1.2 With the machine's process capabilities in its possession, the process program editor may proceed with creating the desired process program. At the conclusion of the editing session, the new machine process program will either be saved at the host or sent directly to the machine for storage and/or use. At this point, the process program is known to satisfy a number of constraints, but it is not necessarily completely acceptable to the machine due to interrelationships of the process program data which are too complex to be described in the machine process capabilities data. The host at any time may verify that a process program is truly valid by sending the process program to the machine and asking it to check the process program and tell the host whether or not the process program is, in fact, correct. If not correct, the equipment is expected to provide information on what data in the process program is unacceptable. This action is accomplished through S7,F31–32. This transaction is equivalent to S7,F23–24, with one important exception, the machine is not to do anything with the process program received under S7,F31 except acknowledge that it got the message (S7,F32) and, as soon as it is able, respond with S7,F27, which provides the host with information on the validity of the process program. In this way, a new version of a process program already held by a machine may be checked for validity without affecting the operation of the machine (i.e., a newer version of a particular process program may be checked while an older version is simultaneously being used by the equipment for material processing).

R1-4.2.1.3 At some point, a host resident process program will be required by the equipment for material processing. Transfer of a program may be accomplished in either of two ways. First, the host may initiate transfer by transmitting S7,F23. In this case, immediately upon reception of the message, the equipment is required to respond with S7,F24, which tells the host that the process program arrived and whether or not the process program is accepted for further processing by the equipment. The second means is for the equipment to initiate the transfer by asking for a process program using S7,F25. In this case, the host will send the process program to the equipment or tell the equipment it is unable to satisfy the request. S7,F23 may also be used by a piece of equipment to transfer a process program to its host for archiving. In this case, the host will respond with S7,F24 and an appropriate completion code. Likewise, a host may request a process program transfer from its machine using S7,F25. The machine will respond with S7,F26, which will contain the process program or an error indication.

R1-4.2.1.4 Following reception by the equipment of the process program, it is the machine's responsibility to check the contents of the process program for validity and respond to the host with a S7,F27 message formatted with the appropriate information about the just received process program. To complete the process program exchange transaction, the host will acknowledge the S7,F27 message with S7,F28. What is done with the process program once accepted and checked for validity is dependent on the state of the process equipment.

#### R1-4.3 *Equipment Process Capabilities Data*

R1-4.3.1 The underlying assumption of SECS-II formatted process programs is that processing instructions for equipment can be expressed as sequences of commands with parameters. Commands are integer codes which tell the machine what to do. The parameters of each command are numeric (integer or floating point) values, Boolean values, or text strings which specify how to carry out the particular command. This provides a very flexible structure for building process programs but does not provide the specific information (code values, types and number of parameters, legal parameter values, etc.) required by a host system to generate a process program for a particular piece of equipment. Under SECS-II, this information is provided to a host via the machine's Equipment Process Capabilities Data or PCD.

R1-4.3.2 A PCD provides three levels of information global data pertaining to the entire process program; definition of each possible command understood by the machine; and definition of each command parameter. Global process program definition data consist of MDLN, SOFTREV, CMDMAX, BYTMAX, and the list of command descriptors.

R1-4.3.3 MDLN and SOFTREV provide the same data to the host as the equipment's response to the S1,F1 host interrogative, "Are you there?". They are included in the PCD to provide a means of distinguishing between PCDs for different machines and revisions of PCDs for the same piece of equipment. Also, when a process program is generated, the MDLN/SOFTREV values of the PCD are provided in the process program to allow the machine an unambiguous method of determining if the process program was generated from a PCD it understands.

R1-4.3.4 BYTMAX and CMDMAX are two integer values which allow the equipment to limit the size of the process program which will be generated. BYTMAX specifies the maximum number of bytes a process program may occupy. CMDMAX specifies the maximum number of commands which may appear in the process program. Either value may be zero, which indicates that no maximum limit is being imposed by the equipment.

R1-4.3.5 The PCD command list identifies (in no particular order) each of the unique operations its associated machine is capable of performing. These operations may correspond to processing operations of the equipment (bake, spin), initialization of equipment components (set beamline controls), definition of data values referenced by later commands (define bond coordinates or inspection points), or even "pseudo-operations," which allow conditional execution of the process program (go to X; if temperature out of range, then go to y; repeat ramping until speed 200; etc.).

R1-4.3.6 Each command in the PCDlist has a number of data values associated with it which provide the host with the command's personality. These are CCODE, CNAME, RQCMD, BLKDEF, BCDS, IBCDS, NBCDS, ACDS, IACDS, NACDS, and the commands parameter list.

R1-4.3.7 CCODE defines the unique numeric code which the equipment recognizes as representing the command being defined. CNAME is a text string which hopefully describes the function of the command. The string must be unique for each command since humans generating process programs at the host will use them, and the host process program generator will translate the CNAME to the corresponding CCODE.





R1-4.3.8 RQCMD. This Boolean value allows equipment to specify whether or not a command must appear at least once within their process program. If true, the command must be used. If RQCMD is set false, the command may or may not be used in the process program at the discretion of the person creating the process program.

R1-4.3.9 In addition to the information the PCD provides on allowed data content within a process program, it also can provide information to the host on possible interdependencies between the commands. Specifically, through the PCD the host can know such things as: command code A must appear before command code B; command code A must come after command code D; command code A must immediately precede command code X; command code A must not come before command code E; command code A must not come after command code F; and/or command code A must immediately come after command code T. Each of the PCD entries, BCDS (before codes), ACDS (after codes), IBCDS (immediately before codes), IACDS (immediately after codes), NBCDS (not before codes), and NACDS (not after codes), is a SECS item which may contain one or more command codes. Each particular item defines the relation to be satisfied. The elements of the item identify the command codes which are to satisfy the relation with the command being defined. A zero length item indicates that no restrictions apply for that type of checking. For example, if the values of the various fields take on the values shown in Figure R1-1, the host process program editor will assure that the TEST command (code 10) will occur before commands with codes 5, 6, and 8; that it will come after commands with codes 100 and 2; that TEST will not appear after the command with code 20; and that each occurrence of command code 3 will have a TEST command immediately before it, subject to the block checking limitations described elsewhere.

CNAME = TEST

CCODE = 10

BCDS = 5,6,8

IBCDS = 3

NBCDS = none

ACDS = 100.2

IACDS = none

NACDS = 20

R1-4.3.10 Associated with before/after checking is the concept of a block which allows setting of limits on before/after checking. A block consists of a start block command, a block terminator command and possibly body commands, commands which are included between the start and terminator commands. There are no specific command codes for start, or terminator commands in SECS-II formatted process programs. Instead, being a start block, terminator block, or body command is merely an attribute of each command defined in the PCD. The field BLKDEF defines this attribute for each command. A positive one indicates the command starts a new block. Zero indicates that the command is a body command and neither starts nor terminates a block. A value of negative one indicates that the command is a terminator command.

R1-4.3.11 Before/after checking for a particular command is performed only with other commands within the same block. To be within the same block, a command must have the same nesting level as the command of interest or the command must be a contained block.

R1-4.3.11.1 The example data in Figure R1-2 shows six grouping of commands for before/after checking: (A,B',N), (B,C,D',G',M), (D,E,F), (G,H',L), (H,I',K), (I,J). A letter followed by an apostrophe (') indicates a block which has been collapsed to a command and has the before/after attributes of its start block command. Note that body and terminator commands occur in only one grouping, while block start commands occur in two. Also, note that the outermost block is assumed to begin with the first command of the process program and to end with the last command.

R1-4.3.12 Each command's parameter list defines the parameters required by the equipment to carry out each particular command. The order in which each parameter descriptor appears in the PCD parameter list also defines

the order in which parameters will appear in a process program command parameter list. Each parameter is one of three possible types: numeric, text, or Boolean.

R1-4.3.13 Regardless of parameter type, the first four elements of any parameter descriptor list are the same. The first field, PNAME, specifies the text string which names the parameter. This data will be displayed by a host when prompting a human for the parameter data. The second field, RQPAR, specifies if the value must be specified at the time the process program is generated (true) or if specifying the data is optional (false). The third field, PDFLT, identifies the type of data to be accepted for this parameter as well as providing default values to include in the process program if the RQPAR is false and no data is input for the parameter when the process program is generated. PDFLT will have zero length if no default value is provided.

Nesting Level	Command Sequence	BLKDEF Value
0 +	A	0
1 +	B	+1
1 +	C	0
2 +	D	+1
2 +	E	0
2 +	F	-1
2 +	G	+1
3 +	H	+1
4 +	I	+1
4 +	J	-1
3 +	K	-1
2 +	L	-1
0 + A 1 0 +	M	-1

R1-4.3.14 The final field, PMAX, specifies the maximum length of the parameter data placed in a process program. For numeric and Boolean data, it specifies the maximum number of data entries in the SECS-II item. For a string parameter, it specifies the maximum number of characters acceptable to the machine. In either case, negative values are invalid and a value of zero indicates there is no length restriction.

R1-4.3.15 For numeric and Boolean parameters which are multi-valued items, usage of PDFLT becomes a bit more complex. In these cases, PDFLT may also be a vector of values. When default values are to be included in a process program, the entry of the default vector in the same ordinal position as the parameter entry requiring the default is used. If the parameter is allowed to have N entries but only M defaults are provided, the last N-M parameter entries will have no defaults.

R1-4.3.16 If the numeric or Boolean vector parameter is required to be entered, PDFLT will contain no default data values, but dummy values must be provided so that the length of the item specifies the minimum number of entries the equipment expects to receive for the parameter. If this minimum number of entries exceeds the maximum number of entries allowed for the parameter (PMAX), then only PMAX entries will be provided in the process program.

R1-4.3.17 Numeric parameters may be any of the SECS recognized floating point or integer data types. PDFLT identifies the particular type. ULIM and LLIM will be of the same data type and specify the range of legal values for the parameter ( $LLIM \leq x \leq ULIM$ ). UNITS is a character string formed according to SEMI E5, § 12, which specifies the expected units of measure of the numeric value. RESC specifies whether the resolution of the data item to be entered is to be in terms of a fundamental increment or significant digits. In the case of the former, RESV will be of the same type as the expected parameter and will specify the base increment. In the case of the latter, RESV will be an integer and will specify the number of significant digits to accept for the parameter.

R1-4.3.18 In addition to the standard fields described above, string parameter descriptors have one unique field. This field provides a set of template strings. A text parameter will be assumed valid by a host process program generator if it matches one of the template strings. A match occurs if the input string is at least as long as the

corresponding template and each position of the template and data strings match. A null string specified as a template will result in a match with any data string. A null data string will match only a null template string. A null template list indicates all strings are acceptable to the equipment.

#### *R1-4.4 Equipment Capabilities Descriptor Availability*

R1-4.4.1 Ideally, each piece of equipment should be able to respond to a host PCD request at any time. However, inasmuch as an equipment's PCD may be rather large and the equipment may have limited storage capacity, constant availability may be impossible. In these cases, some compromise will have to be made such as making it available only at machine initialization or when idle. In extremely severe cases, an equipment manufacturer may have to provide the PCD data with the rest of the machine documentation, requiring his customer to manually enter the data into his host system.

R1-4.4.2 In light of this difficulty, host systems should maintain copies of PCDs for each piece of equipment under its control and not expect to be able to obtain the PCD from the machine whenever it is required. Doing so, in fact, will permit more flexible process program development in the host, allowing creation of process programs even when equipment is not online and encourage the use of formatted process programs by equipment manufacturers.

### **R1-5 Suggested Baseline SECS Equipment Implementation**

#### *R1-5.1 Purpose and Scope*

R1-5.1.1 This document provides a recommendation prepared by the Rigid Disk Subcommittee for generating a baseline implementation of the SECS (SEMI Equipment Communication Standard) standards on production process and test equipment. This document is not a tutorial to aid in understanding SECS but rather serves as an introduction to the requirements of SECS, and a brief guide to the selection of SECS messages for equipment. Actual system requirements of many implementations are beyond the scope of this document. The full standards, SEMI E4 — SEMI Equipment Communications Standard I (SECS-I), and SEMI E5 — SEMI Equipment Communications Standard II (SECS-II) should be consulted by all users.

#### *R1-5.2 Introduction*

R1-5.2.1 The SECS standards are an existing and developed set of communication standards currently used by the semiconductor and other industries to support automated production. The standards provide a means for communicating information and control between production equipment and a "host" computer. This transfer of information and control can be used to provide production tracking and location of WIP (Work-In-Process), scheduling of WIP and control of material transfer at the equipment. The process measurements and records can be used to provide process engineers with a database for statistical process control. SECS messages are appropriate to a wide variety of applications, including measurement, processing, and material transport equipment.

#### *R1-5.3 SECS-I Standard*

R1-5.3.1 SECS-I defines the lower protocol layers of a point-to-point interface between equipment and a host computer system. The standard requires a simple, well understood physical interface, RS-232. The SECS-I protocol allows the equipment control over the protocol: the equipment is the master of the link and can initiate the transfer of a message to the host. Likewise, the equipment can regulate the receipt of a message by its response to a handshake to receive the message. Thus, the interface takes place at the convenience of the equipment, and equipment with very limited computer resources can still support the standard.

#### *R1-5.4 SECS-II Standard*

R1-5.4.1 SECS-II defines the higher layer in the protocol, including message content, structure, and data types and their formats. SECS-II defines messages in sets with related functions called Streams. The actual content of the messages are specific to an application, but it is possible for a properly designed host software system to unpack a message and present the data in a meaningful way with no prior definition as to the content. Equipment need only implement those messages appropriate to meet its system requirements; thus, very simple equipment will require implementation of few messages.

R1-5.4.2 Application Note A2 contains some suggestions for message utilization, including some information on minimum message sets. It is the intent of this report to expand in a somewhat different direction, to identify those messages which typically constitute a sufficient set given a selected equipment function.

R1-5.4.3 To select a message set, the requirements for the equipment must be identified. This requirement, in turn, determines a message set. Identified below are a number of types of equipment requirements and a baseline set of messages supporting those requirements. The message sets identified are baseline recommendations. Actual equipment implementations may need more messages than those specified here to satisfy all system requirements. The published standards should be consulted for all applications. Issues such as handling of multi-block messages, optional replies, and others are described in detail in the SEMI specifications and are not a topic of this baseline recommendation.

R1-5.4.4 The implementation for a specific equipment type begins by specifying which tasks that equipment is required to perform from the following list, and then studying the expanded descriptions for those tasks chosen in the SECS-II implementation section.

Typical Tasks for Measurement and Process:

1. Measurement or Action Reports
2. Equipment Alarm Reports
3. Remote Request for Equipment Condition or State
4. Operator Interface to the Host
5. Remote Access to Process Programs
6. Remote Commands

Typical Tasks for Material Control and Transport:

1. Material Status Information
2. Material Transport Control

Additional Tasks for Special Situations:

1. File Transfer

#### R1-5.5 *Baseline SECS Implementation Recommendations* — SECS-I

R1-5.5.1 The baseline SECS-I requirement for equipment is to fully implement the SECS-I protocol as defined in SEMI E4. This requirement applies to all SECS compatible equipment independent of the equipment's function. The flow chart (Figure 2 of SEMI E4) illustrates the block transfer protocol of SECS-I. The body of the SECS-I standard describes protocol timeouts and other requirements that are essential components of the specification.

#### R1-5.6 *Baseline SECS Implementation Recommendations* — SECS-II

R1-5.6.1 The SECS-II implementation begins with a choice or selection of messages for the equipment. In order for equipment to meet the baseline requirements for a viable SECS-II interface, the equipment must be capable of generating a certain set of messages and recognizing another set. The messages required for either set will depend on tasks of equipment and on system requirements for production and process control by the host. Equipment must accept S1,F1 and send S1,F2. Note that implementation in the reverse direction is optional. In order to ensure a viable data communication link, certain messages are required:

*Messages for All Equipment* — Required by SECS-II

- a. Messages Generated by the Equipment
  - S1,F2: On Line Data
  - S9,F1: Unrecognized Device ID
  - S9,F3: Unrecognized Stream Type
  - S9,F5: Unrecognized Function Type
  - S9,F7: Illegal Data
- b. Messages Recognized by the Equipment
  - S1,F1: Are You There Request

R1-5.6.2 In addition to those messages which are required, the following are strongly recommended. These messages are used for diagnostic purposes:

## *Messages for All Equipment — Strongly Recommended*

### A. Messages Generated by the Equipment

#### 1. S2,F25: String Diagnostic Request

### B. Messages Recognized by the Equipment

#### 1. S2,F26: String Diagnostic Data

R1-5.6.3 Note that messages are identified by a “stream” number and a “function” number. All primary messages have an odd-numbered function, and the corresponding reply is the next consecutive even-numbered function. Thus, many messages are paired. For example, the Stream 1, Function 1 (S1,F1) message generated by either host of equipment has the reply of Stream 1, Function 2 (S1,F2).

R1-5.6.4 *Reports of Measurements or Process Actions* — The simplest task for measurement and process equipment is to report their measurements or actions to the host. This report could also include equipment setup parameters and sample identification. The ability to accurately transfer this data from equipment to a factory host is of prime importance in automating production. Depending on the equipment requirements, the function can be far greater.

R1-5.6.4.1 *Measurements or Action Reports* — Given an equipment requirement to relay data to a host computer, the equipment must handle one or both of the following messages as defined by SEMI E5.

S6,F3: Discrete Variable Send

S6,F9: Formatted Variable Send

R1-5.6.4.2 *Equipment Alarm Reports* — These messages are unsolicited by the host, they transmit information warning of conditions threatening personal safety, equipment safety, or out of limit equipment parameters which may cause harm to the product or indicate equipment malfunction:

S5,F1: Alarm Report Send

R1-5.6.4.3 *Remote Request for Equipment Condition or State* — This is the method where the host requests data from the equipment.

Messages Transmitted to the Equipment:

S6,F7: Data Transfer Request

Messages Transmitted by the Equipment in Reply:

S6,F8: Data Transfer Data

R1-5.6.4.4 *Operator Interface to the Host* — Many equipment systems will have an interactive operator’s console; some will have computer terminals used for this purpose. SECS-II has a message type to transfer text from host to the equipment console. There are also messages through which equipment operators may transmit text from their console, directly to the host. This text may include desired information which is not accessible by the equipment computer directly.

Message Transmitted by the Equipment:

S10,F1: Terminal Request

Message Recognized by the Equipment:

S10,F3: Terminal Display, Single

R1-5.6.4.5 *Remote Access to Process Programs* — SECS-II defines a means for storing or retrieving equipment process programs. This function allows upload and download of such programs through the SECS interface. By this means, programs for equipment may be archived in the host computer system. Either the host or the equipment may request the transfer of a Process Program from the other. To do this:

The Requestor transmits:

S7,F5: Process Program Request

The Sender of the Process Program replies:

S7,F6: Process Program Data

In addition, either host or equipment may initiate sending the Process Program. In this case:

The sender transmits:

S7,F1: Process Program Load Inquire

Receiver replies with:

S7,F2: Process Program Load Grant

Sender transmits:

S7,F3: Process Program Send

The receiver answers:

S7,F4: Process Program Acknowledge

R1-5.6.4.6 *Remote Commands* — The host can initiate a command to the equipment in a manner similar to an operator pressing a button.

The host transmits:

S2,F21: Remote Command Send

The equipment replies:

S2,F22: Remote Command Acknowledge

R1-5.6.4.6.1 Execution of the remote command may cause the equipment to transmit a message to the host at a time greater than the reply time required for the S2,F22 message. If this type of reply is desired, the S6,F3: Discrete Variable Data Send may be used to transfer data to the host.

#### R1-5.6.5 *Typical Tasks for Material Control and Transport*

R1-5.6.5.1 *Material Status Information* — The host may query the equipment for material-in-process information. The information is transmitted only as a answer to a host request.

The equipment recognizes:

S3,F1: Material Status Request

The equipment transmits:

S3,F2: Material Status Data

R1-5.6.5.2 *Material Transport Control* — The SECS-II protocol includes the means to affect automated transfer of material from one SECS-compatible device to another. Baseline compatibility requires the equipment to perform a simple material transfer process, an actual implementation may require means for graceful error recovery as well. This recommendation does not include messages to handle error conditions.

Receiving Material:

Equipment Recognizes S4,F1:

Ready to Send Material

Equipment Transmits:

S4,F3: Send Material

S4,F5: Handshake

Sending Material:

Equipment Transmits:

S4,F1: Ready to Send Material



Equipment Recognizes:

S4,F3: Send Material

S4,F5: Handshake Complete

Equipment Recognizes:

S4,F2: RTS Acknowledge

**R1-5.7 Conclusion** — The baseline requirements for equipment using the SECS standards includes all of SECS-I and a limited selection of messages from SECS-II. The choice of SECS-II messages, and data contained therein, is dictated by the equipment and system requirements. The benefits in using the standards are many, including support for growth in equipment function, and standardization needed for effective automation. The results include automated process monitoring and all of the associated benefits.

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

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