



SEMI E37-1018

SPECIFICATION FOR HIGH-SPEED SECS MESSAGE SERVICES (HSMS) GENERIC SERVICES

This Standard was technically approved by the Information & Control Global Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on May 22, 2018. Available at www.semiviews.org and www.semi.org in October 2018; originally published in 1995; previously published April 2013.

1 Purpose

1.1 High-Speed SECS Message Services (HSMS) provides a means for independent manufacturers to produce implementations which can be connected and interoperate without requiring specific knowledge of one another.

1.2 HSMS is intended as an alternative to SEMI E4 (SECS-I) for applications where higher speed communication is needed or when a simple point-to-point topology is insufficient. SEMI E4 (SECS-I) can still be used in applications where these and other attributes of HSMS are not required.

1.3 HSMS is also intended as an alternative to SEMI E13 (SECS Message Services) for applications where TCP/IP is preferred over OSI.

1.4 It is intended that HSMS be supplemented by subsidiary standards which further specify details of its use or impose restrictions on its use in particular application domains.

2 Scope

2.1 HSMS defines a communication interface suitable for the exchange of messages between computers in a semiconductor factory.

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

3 Referenced Standards and Documents

3.1 SEMI Standards and Safety Guidelines

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

SEMI E5 — Specification for SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E173 — Specification for XML SECS-II Message Notation (SMN)

3.2 IETF Documents¹

IETF RFC 791 — Internet Protocol

IETF RFC 792 — Internet Control Message Protocol

IETF RFC 793 — Transmission Control Protocol

IETF RFC 1120 — Requirements for Internet Hosts – Communication Layers

IETF RFC 1340 — Assigned Numbers (Note: This RFC supersedes RFC 820.)

3.3 IEEE POSIX Document²

IEEE POSIX P1003.12 — Protocol Independent Interfaces (PII)

¹ IETF Secretariat, c/o Association Management Solutions, LLC (AMS), 5177 Brandin Court, Fremont, CA 94538, USA; Telephone: +1.510.492.4080, Fax: +1.510.492.4001, <http://www.ietf.org>

² Institute of Electrical and Electronics Engineers, 3 Park Avenue, 17th Floor, New York, NY 10016-5997, USA; Telephone: +1.212.419.7900, Fax: +1.212.752.4929, <http://www.ieee.org>



4 Terminology

4.1 Definitions

4.1.1 *application program interface (API)* — in the case of TCP/IP, a set of programming conventions used by an application program to prepare for or invoke TCP/IP capabilities.

4.1.2 *communication failure* — a failure in the communication link resulting from a transition to the NOT CONNECTED state from the SELECTED state (see § 9).

4.1.3 *confirmed service (HSMS)* — an HSMS service requested by sending a message from the initiator to the responding entity which requires that completion of the service be indicated by a response message from the responding entity to the initiator.

4.1.4 *connection* — a logical linkage established on a TCP/IP LAN between two entities for the purposes of exchanging messages.

4.1.5 *control message* — an HSMS message used for the management of HSMS sessions between two entities.

4.1.6 *data message* — an HSMS message used for communication of application-specific data within an HSMS session. A Data Message can be a Primary Message or a Reply Message.

4.1.7 *entity* — an application program associated with an endpoint of a TCP/IP connection.

4.1.8 *header* — a 10-byte data element preceding every HSMS message.

4.1.9 *initiator (HSMS)* — the entity requesting an HSMS service. The initiator requests the service by sending an appropriate HSMS message.

4.1.10 *Internet Protocol Address (IP Address)* — a logical address which uniquely identifies a particular attachment to a TCP/IP network.

4.1.11 *local entity* — relative to a particular end point of a connection, the local entity is that entity associated with that endpoint.

4.1.12 *local entity-specific* — general qualifier to any procedure, option, issue, or other implementation matter which is not a subject of this Standard and left to the discretion of the individual supplier.

4.1.13 *message* — a complete unit of communication in one direction. An HSMS Message consists of the Message Length, Message Header, and the Message Text. An HSMS Message can be a Data Message or a Control Message.

4.1.14 *message length* — a 4-byte unsigned integer field specifying the length of a message in bytes.

4.1.15 *open transaction* — a transaction in progress.

4.1.16 *port* — an endpoint of a TCP/IP connection whose complete network address is specified by an IP Address and TCP/IP Port number.

4.1.17 *port number* — (or TCP port number). The address of a port within an attachment to a TCP/IP network which can serve as an endpoint of a TCP/IP connection.

4.1.18 *primary message* — an HSMS Data Message with an odd numbered Function. Also, the first message of a data transaction.

4.1.19 *published port* — a TCP/IP IP Address and Port number associated with a particular entity (server) which that entity intends to use for receiving TCP/IP connection requests. An entity's published port must be known by remote entities intending to initiate connections.

4.1.20 *receiver* — the HSMS Entity receiving a message.

4.1.21 *remote entity* — relative to a particular endpoint of a connection, the remote entity is the entity associated with the opposite endpoint of the connection.

4.1.22 *reply* — an HSMS Data Message with an even-numbered function. Also, the appropriate response to a Primary HSMS Data Message.



4.1.23 responding entity (HSMS) — the provider of an HSMS service. The responding entity receives a message from an initiator requesting the service. In the event of a confirmed service, the responding entity indicates completion of the requested service by sending an appropriate HSMS response message to the initiator of the request. In an unconfirmed service, the responding entity does not send a response message.

4.1.24 session — a relationship established between two entities for the purpose of exchanging HSMS messages.

4.1.25 session entity — an entity participating in an HSMS session.

4.1.26 session ID — a 16-bit unsigned integer which identifies a particular session between particular session entities.

4.1.27 stream (TCP/IP) — a sequence of bytes presented at one end of a TCP/IP connection for delivery to the other end. TCP/IP guarantees that the delivered sequence of bytes matches the presented stream. HSMS subdivides a stream into blocks of contiguous bytes-messages.

4.1.28 T3 — reply timeout in the HSMS protocol.

4.1.29 T5 — connect Separation Timeout in the HSMS protocol used to prevent excessive TCP/IP connect activity by providing a minimum time between the breaking, by an entity, of a TCP/IP connection or a failed attempt to establish one, and the attempt, by that same entity, to initiate a new TCP/IP connection.

4.1.30 T6 — control Timeout in the HSMS protocol which defines the maximum time an HSMS control transaction can remain open before a communications failure is considered to have occurred. A transaction is considered open from the time the initiator sends the required request message until the response message is received.

4.1.31 T7 — connection Idle Timeout in the HSMS protocol which defines the maximum amount of time which may transpire between the formation of a TCP/IP connection and the use of that connection for HSMS communications before a communications failure is considered to have occurred.

4.1.32 T8 — network Intercharacter Timeout in the HSMS protocol which defines the maximum amount of time which may transpire between the receipt of any two successive bytes of a complete HSMS message before a communications failure is considered to have occurred.

4.1.33 Transmission Control Protocol/Internet Protocol (TCP/IP) — a method of communications which provides reliable, connection-oriented message exchange between computers within a network.

4.1.34 Transport Level Interface (TLI) — one particular API provided by certain implementations of TCP/IP which provides a transport protocol and operating system independent definition of the use of any Transport Level protocol.

4.1.35 transaction — a Primary Message and its associated Reply message, if required. Also, an HSMS Control Message of the request (.req) type, and its response Control Message (.rsp), if required.

4.1.36 unconfirmed service (HSMS) — an HSMS service requested by sending a message from the initiator to the responding entity which requires no indication of completion from the responding entity.

5 HSMS Overview and State Diagram

5.1 HSMS defines a communication interface suitable for the exchange of messages between computers in a semiconductor factory using a TCP/IP environment. HSMS uses TCP/IP stream support, which provides reliable two way simultaneous transmission of streams of contiguous bytes. It can be used as a replacement for SECS-I communication as well as other more advanced communications environments.

5.2 The procedure for HSMS communications parallels the more familiar SECS-I communications it replaces. The following steps are followed for any communications (HSMS or otherwise):

1. Obtain a communications link between two entities. In SECS-I, this is the RS232 wire physically connecting host and equipment. In HSMS, the link is a TCP/IP connection obtained by the standard TCP/IP connect procedure. Note that the abstract term ‘entity’ is used instead of ‘host’ or ‘equipment.’ This is because, while HSMS is used for SECS-I replacement, it has more general applications as well. In a SECS-I replacement application, the ‘host’ is an ‘entity’ and the ‘equipment’ is an ‘entity.’
2. Establish the application protocol conventions to be used for exchanging data messages between two entities. For SECS-I, this step is implicit in the fact that semiconductor equipment is physically connected on the two ends of the wire: the protocol is SECS-II. In the case of HSMS, the communications link is a dynamically



established TCP/IP connection on a physical link which may be shared with many other TCP/IP connections using protocols other than HSMS or connections using non-TCP/IP protocols. HSMS adds a message exchange (called the Select procedure) which is used to confirm to both entities that the particular TCP/IP connection is to be used exclusively for HSMS communications.

3. Exchange Data. This is the normal intended purpose of the communications link. In both SECS-I and HSMS, the procedure is to exchange SECS-II encoded messages for the control of semiconductor equipment and/or processes. Data exchange normally continues until one or both of the entities are taken off-line for equipment-specific purposes, such as maintenance.
4. Formally end communications. In SECS-I, there is no formal requirement here; the equipment to be taken off-line stops communicating. In HSMS, a message exchange (either the ‘bilateral’ Deselect procedure or the ‘unilateral’ separate procedure) is used for both parties to confirm that the TCP/IP connection is no longer needed for HSMS communications.
5. Break the communications link. In SECS-I, this is done by physically unplugging the host or equipment from the communications cable, which only occurs during repair or physical reconfiguration of the factory network environment. In HSMS, since it uses the dynamic connection environment of TCP/IP, the TCP/IP connection is logically broken via a release or a disconnect procedure without any physical disconnect from the network medium.

5.3 Two additional procedures, of a diagnostic nature, are supported in HSMS, which are generally not required by a simple SECS-I link or a SECS-I direct replacement. These follow:

1. *Linktest* — This procedure provides a simple confirmation of connection integrity.
2. *Reject* — Because HSMS is intended to be extended to protocols other than just SECS-II (by means of subsidiary standards), it is possible that two entities can be connected (due to a configuration error) which use incompatible subsidiary standards. Also, during initial implementation, incorrect message types may be sent, or they may be sent out of order due to software bugs. The reject procedure is used to indicate such an occurrence.

5.4 *HSMS Connection State Diagram* — The HSMS state machine is illustrated in the diagram below. The behavior described in this diagram defines the basic requirements of HSMS: subsidiary standards may further extend these or other states.

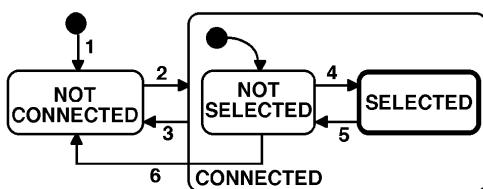


Figure 1
HSMS Connection State Diagram

5.5 State Descriptions

5.5.1 *NOT CONNECTED* — The entity is ready to listen for or initiate TCP/IP connections but either has not yet established any connections or all previously established TCP/IP connections have been terminated.

5.5.2 *CONNECTED* — A TCP/IP connection has been established. This state has two substates, NOT SELECTED and SELECTED.

5.5.2.1 *NOT SELECTED* — A substate of CONNECTED in which no HSMS session has been established or any previously established HSMS session has ended.

5.5.2.2 *SELECTED* — A substate of CONNECTED in which at least one HSMS session has been established. This is the normal ‘operating’ state of HSMS: data messages may be exchanged in this state. It is highlighted by a heavy outline in the state diagram.



5.6 State Transition Table

Table 1 State Transition

#	Current State	Trigger	New State	Actions	Comment
1	...	Local entity-specific preparation for TCP/IP communication.	NOT CONNECTED	Local entity-specific	Action depends on connection procedure to be used: active or passive.
2	NOTCONNECTED	A TCP/IP connection is established for HSMS communication.	CONNECTED - NOT SELECTED	Local entity-specific	None.
3	CONNECTED	Breaking of TCP connection.	NOT CONNECTED	Local entity-specific	See § 6.4.
4	NOT SELECTED	Successful completion of HSMS Select Procedure.	SELECTED	Local entity-specific	HSMS communication is now fully established: data message exchange is permitted.
5	SELECTED	Successful completion of HSMS Deselect or Separate.	NOT SELECTED	Local entity-specific	This transition normally indicates the end of HSMS communication and so an entity would immediately proceed to break the TCP/IP connection (transition 3 above).
6	NOT SELECTED	T7 Connection Timeout.	NOT CONNECTED	Local entity-specific	Per § 9.2.2.

6 Use of TCP/IP

6.1 *TCP/IP API* — The specification of a required TCP Application Program Interface (API) for use in implementations is outside the scope of HSMS. A given HSMS implementation may use any TCP/IP API—sockets, TLI (Transport Layer Interface), etc.—appropriate to the intended hardware and software platform, as long as it provides interoperable TCP/IP streams protocol on the network.

6.1.1 The appendix contains examples of the TCP/IP procedures referenced in this Standard and sample scenarios using both the TLI (POSIX standard 1003.12) and the popular BSD socket model for TCP/IP communication.

6.2 TCP/IP Network Addressing Conventions

6.2.1 *IP Addresses* — Each physical TCP/IP connection to a given Local Area Network (LAN) must have a unique IP Address. IP Addresses must be assignable at installation time, and an HSMS implementation cannot select a fixed IP Address. A typical IP Address is 192.9.200.1.

6.2.2 IP imposes restrictions on these numbers which are outside the scope of the HSMS protocol. Consult § 2.3 of RFC 791, Internet Protocol (IP) in § 3.

6.2.3 *TCP Port Numbers* — A TCP Port Number can be considered as an extension of the IP Address.

6.2.4 HSMS implementations should allow configuring TCP Port to the full range of the TCP/IP implementation used. A typical TCP Port Number is 5000.

6.2.5 Conventions have been established for selecting TCP Port Numbers which are outside the scope of the HSMS protocol. Consult RFC 793, Transmission Control Protocol (TCP) in § 3.

6.3 Establishing a TCP/IP Connection

6.3.1 *Connect Modes* — The procedures for establishing a TCP/IP connection are defined in RFC 793. However, not all the procedures defined by RFC 793 are supported by commonly available APIs. In particular, while RFC 793 permits both entities to initiate the connection simultaneously, this feature is rarely supported in available.



6.3.2 APIs. Therefore, HSMS restricts an entity to one of the following modes:

- *Passive Mode* — The Passive mode is used when the local entity listens for and accepts a connect procedure initiated by the Remote Entity.
- *Active Mode* — The Active mode is used when the connect procedure is initiated by the Local Entity.

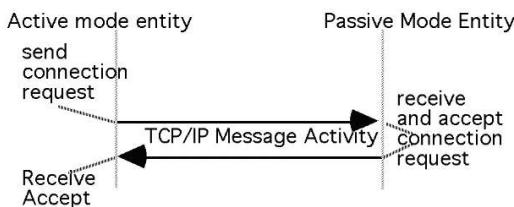


Figure 2
TCP/IP Connection Procedure Summary

6.3.3 The appendix provides an example of how an entity may operate alternately in the active and passive modes to achieve greater flexibility in establishing communications.

6.3.4 *Passive Mode Connect Procedure* — The procedure followed by the Passive Local Entity is defined in RFC 793. It is summarized as follows:

1. Obtain a connection endpoint and bind it to a published port.
2. Listen for an incoming connect request to the published port from a remote entity.
3. Upon receipt of a connect request, acknowledge it and indicate acceptance of the connection. At this point, the connect procedure has completed successfully, and the CONNECTED state is entered (see § 5).

6.3.5 These procedures are carried out through the API of the local entity's implementation of TCP/IP. The appendices provide the API-specific procedures for the above steps using both TLI and BSD.

NOTE 1: A failure may occur during the above steps. The reason for failure may be local entity-specific or may be due to a lack of any connect request after a local entity-specific timeout. The action to be taken (e.g., return to step 1 to retry) is a local entity-specific issue.

NOTE 2: See § 9, Special Considerations, for issues relating to multiple connection requests to the same passive mode entity.

6.3.6 *Active Mode Connect Procedure* — The procedure followed by the Active Local Entity is defined in RFC 793. It is summarized as follows:

1. Obtain a connection endpoint.
2. Initiate a connection to the published port of a passive mode remote entity.
3. Wait for the receipt of the acknowledgement and the acceptance of the connect request from the remote entity. Receipt of the acceptance from the remote entity indicates successful completion of the connect procedure, and the CONNECTED state is entered (see § 5).

6.3.7 These procedures are carried out through the API of the local entity's implementation of TCP/IP. The appendix provides the API-specific procedures for the above steps using both TLI and BSD.

NOTE 3: A failure may occur during the above steps. The reason for failure may be local entity-specific or may be due to a lack of any accept message after a local entity-specific timeout. The action to be taken is a local entity-specific issue. If, however, the local entity intends to retry the connection, it should do so subject to the T5 connect separation timeout (see 'Special Considerations').

6.4 *Terminating a TCP/IP Connection* — Connection termination is the logical inverse of Connection establishment. From the Local Entity's perspective, a TCP/IP connection may be broken at any time. However, HSMS only permits termination of the connection when the connection is in the NOT SELECTED substate of the CONNECTED state.



6.5 The procedures for termination of a connection are defined in RFC 793. Either entity may initiate termination of the connection. The NOT CONNECTED state is entered, indicating the end of HSMS communications. The appendix illustrates the procedures for both release and disconnect using the TLI and BSD APIs.

7 HSMS Message Exchange Procedures

7.1 HSMS defines the procedures for all message exchange between entities across the TCP/IP connection established according to the procedures in the previous section. As explained in the overview, once the connection is established, the two entities establish HSMS communications with the Select procedure. Then data messages may be exchanged in either direction at any time. When the entities wish to end HSMS communications, the Deselect or Separate procedure is used to end HSMS communications.

7.2 *Sending and Receiving HSMS Messages* — All HSMS procedures involve the exchange of HSMS messages. These messages are sent and received as TCP/IP streams using the previously established TCP/IP connection at standard priority. In particular, the use of ‘Urgent’ data is not supported under HSMS (see RFC 793 for more information on send and receive procedures).

7.2.1 The appendix gives examples of sending and receiving HSMS messages using both TLI and BSD socket APIs.

7.3 *Select Procedure* — The Select procedure is used to establish HSMS communications on a TCP/IP connection using the Select.req and Select.rsp messages in a control transaction.

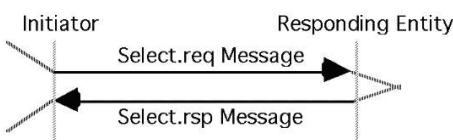


Figure 3
Select Control Transaction

7.4 Although HSMS permits Select at any time in the CONNECTED state, subsidiary standards may further require the connection to be in the NOT SELECTED substate (see ‘Special Considerations’).

7.4.1 *Initiator Procedure* — The procedure followed by the initiator is as follows.

1. The initiator of the select procedure sends the Select.req message to the responding entity.
2. If the initiator receives a Select.rsp with a Select Status of 0, The HSMS Select procedure completes successfully and the SELECTED state is entered (see § 5).
3. If the initiator receives a Select.rsp with a non-zero Select Status, the Select completes unsuccessfully (no state transitions).
4. If the T6 timeout expires in the initiator before receipt of a Select.rsp, it is considered a communications failure (see ‘Special Considerations’).

7.4.2 *Responding Entity Procedure* — The procedure followed by the responding entity is as follows.

1. The responding entity receives the Select.req.
2. If the responding entity is able to accept the select, it transmits the Select.rsp with a Select Status of 0. The HSMS Select Procedure for the responding entity is successfully completed, and the SELECTED state is entered (see § 5).
3. If the responding entity is unable to permit the select, it transmits the Select.rsp with a nonzero Select Status. The HSMS Select Procedure for the responding entity completes unsuccessfully (no state transitions).

7.4.3 *Simultaneous Select Procedures* — If the subsidiary standards do not restrict the use of the Select, it is possible that both entities simultaneously initiate Select Procedures with identical SessionID’s. In such a case, each entity shall accept the other entity’s select request by responding with a Select.rsp.

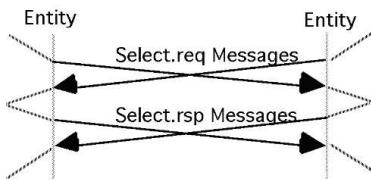


Figure 4
Simultaneous Select Transactions

7.5 Data Procedure — HSMS data messages may be initiated by either entity as long as the connection is in the SELECTED state. Receipt of a data message when not in the SELECTED state shall result in a reject procedure (see § 7.7).

7.6 Data messages may be further defined as part of a data transaction as either a ‘Primary’ or ‘Reply’ data message. In a data transaction, the initiator of the transaction sends a primary message to the responding entity. If the Primary message indicates that a reply is expected, a Reply message is sent by the responding entity in response to the Primary.

7.6.1 The following types of Data Transactions are supported:

1. Primary Message with reply expected and the associated Reply Message.
2. Primary Message with no reply expected.

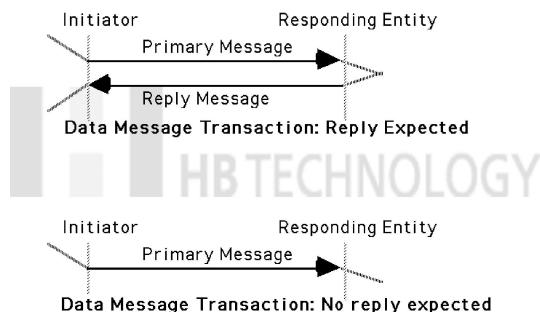


Figure 5
Data Message Transaction

7.6.2 The specific procedures for these transactions are determined by the application layer and are subject to other Standards (e.g., SEMI E5 and SEMI E30 for GEM equipment using SECS-II encoded messages).

7.6.3 The applicable upper layer standard is identified by the message type. The type is determined from the specific format defined in § 8. The normal type for HSMS messages is SECS-II text. Also refer to ‘Special Considerations’ concerning the T3 Reply Timeout.

7.7 Deselect Procedure — The Deselect procedure is used to provide a graceful end to HSMS communication for an entity prior to breaking the TCP/IP connection. HSMS requires that the connection be in the SELECTED state. The procedure is as follows.

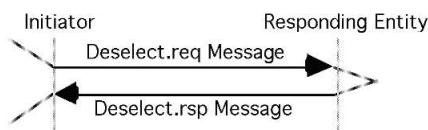


Figure 6
Deselect Control Transaction



7.7.1 Initiator Procedure

1. The initiator of the Deselect procedure sends the Deselect.req message to the responding entity.
2. If the initiator receives a Deselect.rsp with a Deselect Status of 0, its Deselect procedure terminates successfully. The NOT SELECTED state is entered (see § 5).
3. If the initiator receives a Deselect.rsp with a non-zero Deselect Status, its Deselect procedure terminates unsuccessfully. No state change occurs.
4. If the T6 timeout expires in the initiator before receipt of a Deselect.rsp, it is considered a communications failure (see ‘Special Considerations’).

7.7.2 Responding Entity Procedure

1. The responding entity receives the Deselect.req message.
2. If the responding entity is in the SELECTED state, and if it is able to permit the Deselect, it responds using the Deselect.rsp with a zero response code. The responding entity's Deselect procedure completes successfully. The NOT SELECTED state is entered (see § 5).
3. If the responding entity is unable to permit the Deselect, either because it is not in the SELECTED state or because local conditions do not permit the Deselect, it responds using the Deselect.rsp with a non-zero response code. The responding entity's Deselect procedure terminates unsuccessfully. No state change occurs.

7.7.3 Simultaneous Deselect Procedures — If the subsidiary standards do not restrict the use of the Deselect, it is possible that both entities simultaneously initiate Deselect Procedures with identical SessionID’s. In such a case, each entity shall accept the other entity’s Deselect request by responding with the deselect.rsp.

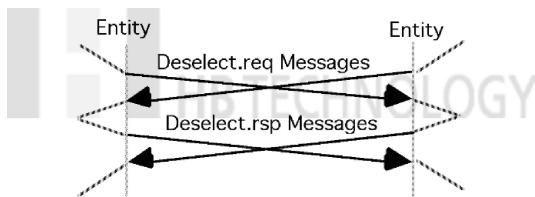


Figure 7
Simultaneous Deselect Transactions

7.8 Linktest Procedure — The Linktest is used to determine the operational integrity of TCP/IP and HSMS communications. Its use is valid anytime in the CONNECTED state.

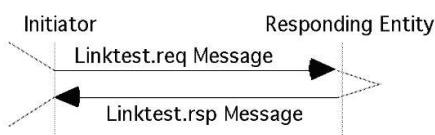


Figure 8
Linktest Control Transaction

7.8.1 Initiator Procedure

1. The initiator of the Linktest procedure sends the Linktest.req message to the responding entity.
2. If the initiator receives a Linktest.rsp within the T6 timeout, the Linktest is successfully completed.
3. If the T6 timeout expires in the initiator before receipt of a Linktest.rsp, it is considered a communications failure (see ‘Special Considerations’).



7.8.2 Responding Entity Procedure

1. The responding entity receives the Linktest.req from the initiator.
2. The responding entity sends a Linktest.rsp.

7.9 Separate Procedure — The Separate procedure is used to abruptly terminate HSMS communication for an entity prior to breaking the TCP/IP Connection. HSMS requires that the connection be in the SELECTED state when using Separate. The responding entity does not send a response and is required to terminate communications regardless of its local state. The procedure is as follows.



Figure 9
Separate Control Transaction

7.9.1 Initiator Procedure

1. The initiator of the select procedure sends the Separate.req message to the responding entity. The initiator's Separate procedure completes successfully. The NOT SELECTED state is entered (see § 5).

7.9.2 Responding Entity Procedure

1. The responding entity receives the Separate.req from the initiator.
2. If the responding entity is in the SELECTED state, its Separate procedure completes successfully.
3. If the responding entity is not in the SELECTED state, the Separate.req is ignored.

7.10 Reject Procedure — The Reject procedure is used in response to an otherwise valid HSMS message received in an inappropriate context. Supporting the reject procedure can provide useful diagnostic information during the development of a distributed application using HSMS. The procedure is as follows:

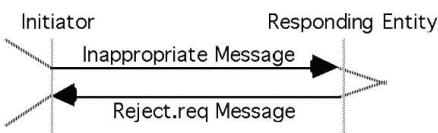


Figure 10
Reject Transaction

7.10.1 Initiator (Sender of Inappropriate Message) Procedure

1. The initiator of the inappropriate message, upon receiving the Reject.req, takes appropriate action (local entity-specific).

7.10.2 Responding Entity Procedure

1. The entity receiving the inappropriate message responds with a Reject.req message.

7.10.3 HSMS requires the reject procedure for the receipt of a data message in the NOT SELECTED state, or the receipt of a message whose SType or PType (see next section: Message Format) is not defined for the entity receiving the message. Subsidiary standards may define other conditions which require the Reject Procedure. In general, receipt of a reject message is an indication of an improperly configured system or a software programming error.

8 HSMS Message Format

8.1 This section defines the detailed format of the messages used by the procedures in the previous section.



8.2 General Message Format

8.2.1 Byte Structure — Within HSMS, a byte contains eight (8) bits. The bits in a byte are numbered from Bit 7 (most significant) to Bit 0 (least significant).

8.2.2 Message Format — An HSMS Message is transmitted as a single contiguous stream of bytes in the following order:

Table 2 HSMS Message Format

Number of Bytes	Description
4 Bytes	Message Length. MSB First. Specifies the number of bytes in the Message Header plus the Message Text.
10 Bytes	Message Header.
0–n Bytes	Message Text. Format is further specified by PType field of message header.

8.2.3 Message Length — Message Length is a four-byte unsigned integer value which specifies the length in bytes of the Message Header plus the Message Text. Message Length is transmitted most significant byte (MSB) first and least significant byte (LSB) last.

8.2.4 The minimum possible Message Length is 10 (Header only). The maximum possible Message Length is implementation-specific.

8.2.5 Message Header — The Message Header is a ten-byte field. The bytes in the header are numbered from byte 0 (first byte transmitted) to byte 9 (last byte transmitted). The format of the Message Header is as follows:

Table 3 HSMS Message Header

Bytes	Description
0–1	Session ID (Device ID)
2	Header Byte 2
3	Header Byte 3
4	PType
5	SType
6–9	System Bytes

8.2.6 The physical byte order is designed to correspond as closely as possible to the SECS-I header.

8.2.6.1 Session ID — Session ID is a 16-bit unsigned integer value, which occupies bytes 0 and 1 of the header (byte 0 is MSB, 1 is LSB). Its purpose is to provide an association by reference between control messages (particularly Select and Deselect) and subsequent data messages. It is the role of HSMS subsidiary standards to specify this association further.

8.2.6.2 Header Byte 2 — This header byte is used in different ways for different HSMS messages. For Control Messages (see SType, below) it contains zero or a status code. For a Data Message whose PType (see below) = 0, it contains the W-Bit and SECS Stream. For a Data Message with PType not equal to 0, see ‘Special Considerations.’

8.2.6.3 Header Byte 3 — This header byte is used in different ways for different HSMS messages. For Control Messages, it contains zero or a status code. For a Data Message whose PType (see below) = 0, it contains the SECS Function. For a Data Message with PType not equal to 0, see ‘Special Considerations.’

8.2.6.4 PType — PType (Presentation Type) is an 8-bit unsigned integer value which occupies byte 4 of the header. PType is intended as an enumerated type defining the presentation layer message type: how the Message Header and Message Text are encoded. Only PType = 0 is defined by HSMS to mean SECS-II message encoding. For non-zero PType values, see ‘Special Considerations.’

**Table 4 PType**

<i>Value</i>	<i>Description</i>
0	SECS-II Encoding
1–127	Reserved for subsidiary standards
128–255	Reserved, not used

8.2.6.5 *SType* — *SType* (Session Type) is a one-byte unsigned integer value which occupies header byte 5.

8.2.6.6 *SType* is an enumerated type identifying whether this message is an HSMS Data Message (value = 0) or one of the HSMS Control Messages (other). Those values not explicitly defined in the table are addressed in ‘Special Considerations.’

Table 5 SType

<i>Value</i>	<i>Description</i>
0	Data Message
1	Select.req
2	Select.rsp
3	Deselect.req
4	Deselect.rsp
5	Linktest.req
6	Linktest.rsp
7	Reject.req
8	(not used)
9	Separate.req
10	(not used)
11–127	Reserved for subsidiary standards
128–255	Reserved, not used

8.2.6.7 *System Bytes* — System Bytes is a four-byte field occupying header bytes 6–9. System Bytes is used to identify a transaction uniquely among the set of open transactions.

8.2.6.8 *Uniqueness* — The System Bytes of a Primary Data Message, Select.req, Deselect.req, or Linktest.req message must be unique from those of all other currently open transactions initiated from the same end of the connection. They must also be unique from those of the most recently completed transaction.

8.2.6.9 *Reply Message* — The System Bytes of a Reply Data Message must be the same as those of the corresponding Primary Message. The System Bytes of a Select.rsp, Deselect.rsp, or Linktest.rsp must be the same as those of the respective ‘.req’ message.

8.3 *HSMS Message Formats by Type* — The specific interpretation of the header bytes in an HSMS message is dependent on the specific HSMS message type as defined by the value of the *SType* field. The complete set of messages defined is summarized in the table below, shown for *PType* = 0 (SECS-II message format).

**Table 6 HSMS Message Format Summary**

Message Type	Message Header						Message Text
	Bytes 0–1 SessionID	Byte 2	Byte 3	Byte 4 PType	Byte 5 SType	Bytes 6–9 System Bytes	
Data Message *		W-Bit and SECS Stream	SECS Function	0	0	Primary: UniqueReply: Same as primary	Text
Select.req	*	0	0	0	1	Unique	none
Select.rsp	Same as .req	0	Select Status	0	2	Same as .req	none
Deselect.req	*	0	0	0	3	Unique	none
Deselect.rsp	Same as .req	0	Deselect Status	0	4	Same as .req	none
Linktest.req	0xFFFF	0	0	0	5	Unique	none
Linktest.rsp	0xFFFF	0	0	0	6	Same as .req	none
Reject.req	same as message being rejected	PType or SType of message being rejected	Reason Code	0	7	Same as message being rejected	none
Separate.req	*	0	0	0	9	Unique	none

#1 * Indicates further specification by subsidiary standards.

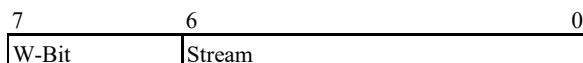
8.3.1 *SType=0: Data Message* — An HSMS message with SType = 0 is used by the HSMS Data procedure to send a Data message, either Primary or Reply. The message format is as follows:

8.3.2 HSMS Message Length is always 10 (the length of the header alone) or greater.

8.3.3 The HSMS Message Header is as follows:

8.3.3.1 *Session ID* — As described above. Specific value subject to subsidiary standards.

8.3.3.2 *Header Byte 2* — For messages with PType value = 0 (SECS-II), header byte 2 is formatted as shown below.



8.3.3.3 The most significant bit (bit 7) of Header Byte 2 is the W-Bit. In a Primary Message, the W-Bit indicates whether the Primary Message expects a Reply message. A Primary Message which expects a Reply should set the W-Bit to 1. A Primary Message which does not expect a Reply should set the W-Bit to 0. A Reply Message should always set the W-Bit to 0. The low-order 7 bits (bits 6-0) of Header Byte 2 contain the SECS Stream for the message. The Stream is a 7-bit unsigned integer value, which identifies a major topic of the message, and its use is defined within SEMI E5 (SECS-II).

8.3.3.4 *Header Byte 3* — For messages whose PType value = 0, header Byte 3 contains the SECS Function for the message. The Function is an 8-bit unsigned integer value which identifies a minor topic of the message (within the Stream), and its use is defined within SEMI E5 (SECS-II). The least significant bit (bit 0) of the Function defines whether the Data Message is Primary or Reply; the value 1 indicates Primary and the value 0 indicates Reply.

8.3.3.5 *PType* — Set PType = 0 for SECS-II messages.

- SType = 0

8.3.3.6 *System Bytes* — For PType = 0 (SECS-II), the following definition applies. For a Primary Message, System Bytes contain a value uniquely identifying this transaction from all other open transactions initiated from the same end of the Connection. For a Reply Message, System Bytes contain the same value as the corresponding Primary Message.

8.3.3.7 The HSMS Message Text contains the text of the Data Message (if any), formatted as specified by the PType field. For PType = 0, the text shall be formatted as SECS-II messages.



NOTE 4: Some Data Messages consist of header only, with no text.

8.3.4 *SType=1: Select.req* — An HSMS message with SType 1 is a ‘Select Request’ Control Message, which is used by the initiator of the procedure for establishing HSMS communications. The message format is as follows:

8.3.4.1 Message Length is always 10 (Header only).

8.3.4.2 The HSMS Message Header is as follows:

8.3.4.3 *SessionID* — As described above. Specific value subject to subsidiary standards.

- Header Byte 2 = 0
- Header Byte 3 = 0
- PType = 0
- SType = 1

8.3.4.4 *System Bytes* — A unique value among open transactions.

8.3.5 *SType=2: Select.rsp* — An HSMS message with SType 2 is a ‘Select Response’ Control Message, used as the response to a Select.req Control message in the procedure for establishing HSMS communications. The message format is as follows:

8.3.6 Message Length is always 10 (Header only).

8.3.7 The HSMS Message Header is as follows:

8.3.7.1 *SessionID* — Must be equal to the value of the session ID in the corresponding Select.req.

- Header Byte 2 = 0

8.3.7.2 *Header Byte 3* — SelectStatus. A code of zero indicates success of the Select operation. A nonzero code indicates failure.

Table 7 SelectStatus

<i>Value</i>	<i>Description</i>
0	Communication Established. Select was successfully completed.
1	Communication Already Active. A previous select has already established communications to the entity being selected in this select.
2	Connection Not Ready. The Connection is not yet ready to accept select requests.
3	Connect Exhaust. The connection was accepted, but the entity is already servicing a separate TCP/IP connection and is unable to service more than one at any given time.
4–127	Reserved for subsidiary standard-specific reasons for select failure.
128–255	Reserved for local entity-specific reasons for select failure.

- PType = 0
- SType = 2

8.3.7.3 *System Bytes* — Equal to value of System Bytes in the corresponding Select.req.

8.3.8 *SType=3: Deselect.req* — An HSMS message with SType 3 is a ‘Deselect Request’ Control Message, used by the initiator of the Select procedure for ending HSMS communication. The message format is as follows:

8.3.9 Message Length is always 10 (Header only).

8.3.10 The HSMS Message Header is as follows:

8.3.10.1 *SessionID* — The SessionID must match the value of the SessionID of a previously sent Select.req to indicate the particular HSMS session that is ending. Subject to further specification by subsidiary standards.



- Header Byte 2 = 0
- Header Byte 3 = 0
- PType = 0
- SType = 3

8.3.10.2 *System Bytes* — A unique value among open transactions.

8.3.11 *SType=4: Deselect.rsp* — An HSMS message with SType 4 is a ‘Deselect Response’ Control Message, used as the response to a Deselect.req Control message in the Deselect procedure for ending HSMS communications. The message format is as follows:

8.3.12 Message Length is always 10 (Header only).

8.3.13 The HSMS Message Header is as follows:

8.3.13.1 *SessionID* — Must equal the session ID in the corresponding Deselect.req

- Header Byte 2 = 0

8.3.13.2 *Header Byte 3* — DeselectStatus. A code of zero indicates success of the Deselect operation. A nonzero code indicates failure.

Table 8 Deselect Status

<i>Value</i>	<i>Description</i>
0	Communication Ended. The Deselect completed successfully.
1	Communication Not Established. HSMS communications has not yet been established with a select, or has already been ended with a previous Deselect.
2	Communication Busy. The session is still in use by the responding entity and so it cannot yet relinquish it gracefully. In this case, if the original requester must terminate communications, the separate procedure should be used as a last resort.
3–127	Reserved for subsidiary standard-specific reasons for Deselect failure.
128–255	Reserved for local entity-specific reasons for Deselect failure.

- PType = 0
- SType = 4

8.3.13.3 *System Bytes* — Equal to System Bytes in corresponding Deselect.req.

8.3.14 *SType=5: Linktest.req* — An HSMS message with SType 5 is a ‘Linktest Request’ Control Message. It is used to verify the integrity of the HSMS Connection, or as a periodic heartbeat. The message format is as follows:

8.3.15 Message Length is always 10 (Header only).

8.3.16 The HSMS Message Header is as follows:

- SessionID = 0xFFFF (in binary, all ones)
- Header Byte 2 = 0
- Header Byte 3 = 0
- PType = 0
- SType = 5

8.3.16.1 *System Bytes* — A unique value among open transactions.

8.3.17 *SType=6: Linktest.rsp* — An HSMS message with SType 6 is a ‘Linktest Response’ Control Message, used as the response to a Linktest.req Control message in the Linktest Procedure. The message format is as follows:



8.3.18 Message Length is always 10 (Header only).

8.3.19 The HSMS Message Header is as follows:

- SessionID = 0xFFFF (binary, all ones)
- Header Byte 2 = 0
- Header Byte 3 = 0
- PType = 0
- SType = 6

8.3.19.1 *System Bytes* — Equal to System Bytes in corresponding Linktest.req.

8.3.20 *SType=7: Reject.req* — An HSMS message with SType 7 is used in response to any valid HSMS message received which is not supported by the receiver of the message or which is not valid at the time. It is intended for dealing with attempts to use subsidiary standards or user-defined extensions which are not supported by the receiver (e.g., SType equal to any value not defined in this Standard). It must be used when an entity receives a control message which is a response (even numbered SType) for which there was no corresponding open transaction.

8.3.21 The HSMS Message Header is as follows:

8.3.21.1 *SessionID* — Equal to the value of the Session ID in the message being rejected.

8.3.21.2 *Header Byte 2* — For ReasonCode = PType Not Supported, equal to the PType in the message being rejected. Otherwise equal to the value of the SType in the message being rejected.

8.3.21.3 *Header Byte 3* — Reason code (always nonzero)

Table 9 ReasonCode

Value	Description
1	SType Not Supported. A message was received whose SType value not defined in the HSMS standard or the particular subsidiary standard(s) supported by the entity.
2	PType Not Supported. As above, but for PType.
3	Transaction Not Open. A Response control message was received when there was no outstanding request message which corresponded to it.
4	Entity Not Selected. A data message was received when not in the SELECTED state.
4–127	Reserved for subsidiary standard-specific reasons for reject.
128–255	Reserved for local entity-specific reasons for reject.

- PType = 0
- SType = 7

8.3.21.4 *System Bytes* — Equal to System Bytes in corresponding message being rejected.

8.3.22 *SType=9: Separate.req* — An HSMS message with SType = 9 is used to terminate HSMS communications immediately. With the exception of the SType value, it is identical to the Deselect.req message. Its purpose is to end HSMS communications immediately and without exception. No response is defined.

9 Special Considerations

9.1 General Considerations

9.1.1 *Communications Failures* — If a communications failure is detected, the entity should terminate the TCP/IP connection. Upon termination of the connection, the entity may, at this point, attempt to reestablish communications.



9.2 TCP/IP Considerations

9.2.1 Connect Separation Time (T5) — The connect procedures initiate some network activity. Frequent use of the active mode connect procedure to the IP Address and Port Number of an entity not yet ready to accept connections can be hostile to TCP/IP operations. The passive mode does not generate network activity and is not considered hostile to the network, although it may affect local application performance. An Entity initiating a connection in the active mode should limit its use of the connect procedure in a manner that is equivalent to the procedure described here.

9.2.1.1 After an active connect procedure terminates by any means (successfully or unsuccessfully), the Entity should not initiate another active connect procedure (for the same Remote Entity) until the T5 Connect Separation Time has elapsed. The separation of connect operations shall be the sum of the T5 Connect Separation Time interval, plus the duration of the connect operation itself.

9.2.2 NOT SELECTED Timeout (T7) — Entry into the NOT SELECTED state is achieved either by state transition #2 (establishment of a TCP/IP connection). There is a time limit on how long an entity is allowed to remain in the NOT SELECTED state without entering the SELECTED state (transition #4) or returning to the NOT CONNECTED state (transition #6).

9.2.2.1 Some entities, particularly those unable to accept more than a single TCP/IP connection, may be impaired in their operation by remaining in their NOT SELECTED state as they will be unavailable for communications with other entities. Such entities shall disconnect the TCP/IP connection (State Transition Event #3) if communication remains in the NOT SELECTED state for longer than the T7 timeout period.

9.2.3 Network Intercharacter Timeout (T8) — Because TCP/IP is a stream rather than a message protocol, it is possible that bytes which are all part of a single HSMS message may be transmitted in separate TCP/IP messages without any violation of the TCP/IP protocol. Since it is possible that these separate messages may be separated by a substantial period of time, the Network Intercharacter Timeout (T8) is defined.

9.2.3.1 T8 is similar in purpose to the SECS-I T1 timer except that the communications issues which necessitate T8 are not entirely in the control of the sender of the message. Therefore, it is defined only in terms of the receiver of the message. In particular, if after receipt of a partial message, the T8 timeout period expires prior to receipt of the complete message, the receiving entity shall consider such case as a communications failure, as defined above.

9.2.4 Multiple Connection Requests Directed to a Single Published Port — Once a passive entity has accepted a connection on its published port, TCP/IP permits (though does not require) the entity to listen for and accept additional connections directed to the same published port.

9.2.4.1 HSMS permits (though does not require) entities to operate in this manner. However, for the purposes of HSMS compliance, each connection so formed must exhibit the behavior defined in the HSMS state diagram as if it were completely independent of any other connection to the same published port.

9.2.4.1.1 Rejection of Additional Connection Requests by a Passive Mode Entity — A passive mode entity unable to service more than a single TCP/IP connection for HSMS communications shall follow one of these three procedures with respect to additional connection requests.

- Accept the connection, but always respond to any subsequent HSMS select procedures with the Communication Already Active response code. For the purpose of the HSMS State Diagram, the connect procedure terminates successfully (enters CONNECTED state), but HSMS communications are never established (remain in NOT SELECTED substate). This is the preferred option in that it can provide the most information to the remote entity as to why the connection is refused (see HSMS Select Procedure), but places an addition implementation requirement on the local entity.
- Actively reject the connection request. This can be done in a TLI implementation using the t_snddis procedure. This shall cause the connect procedure in the remote entity to terminate unsuccessfully. This option may not be available to all implementations because some API's, notably some implementations of BSD Sockets, do not provide for initiating an active reject. Note, however, that all TCP/IP implementations, including BSD Sockets, properly respond to an active reject from the remote entity.
- Refuse to listen for or accept the connect request. No action is taken in the local entity: the remote entity's connect procedure will eventually time out. This option is permitted, but not recommended, as it can cause considerable



delay on the part of the remote entity. However, it may be the only alternative available to implementations with network resource limitations.

9.2.4.1.2 The documentation of the passive local entity shall indicate which means it uses to refuse connections.

9.3 HSMS-Specific Considerations

9.3.1 *Control Transactions T6 Control Timeout* — A number of the control messages are part of procedures which require a message exchange or transaction: <xx>.req from the initiator of the control service, followed by an <xx>.rsp from the receiver of the <xx>.req in response to it. A control transaction is considered open from the time the <xx>.req request is sent until the time the <xx>.rsp is received.

9.3.1.1 The time a control transaction may remain open is subject to the T6 control transaction timeout. Upon initiation of a control transaction, the local entity should set a timer whose duration is equal to the T6 timeout value. If the transaction is properly closed prior to the expiration of the timer, the timer should be canceled. If the timer expires prior to the proper closing of the transaction, the transaction shall be considered closed by the initiator and considered an HSMS communications failure.

9.3.2 *Procedures and ‘Stateless’ Transactions* — Most of the HSMS control procedures involve a transaction: the initiator sends a request message to the responding entity and waits for a response message. The responding entity receives the initiator's request message and sends a reply.

9.3.2.1 Note that such transactions are ‘stateless’ in the following sense: while the initiator of a transaction is waiting for a response, it may receive a message other than that response, and this message may be any message valid for the state the initiator was in at the time the original transaction was initiated. For example, the two entities may simultaneously initiate transactions. As a result, no states for ‘TRANSACTION OPEN’ or ‘TRANSACTION NOT OPEN’ are reflected in the HSMS state machine. The use of such state information in an implementation is strictly a local entity-specific issue.

9.3.3 *Alternative Message Types and Header Byte Values* — The HSMS standard does not completely define all possible enumerated values of either the PType or SType field. Further, Header bytes 2 and 3 have a format determined by the PType for messages whose SType is equal to 0, but is otherwise specified for all other SType values. The message text formatting is defined by the PType as well, but only for data messages.

9.3.3.1 Subsidiary standards must be consistent with this convention. In particular, for SType = 0, subsidiary standards defining PType values not equal to 0 may specify both the message text encoding and the interpretation of header bytes 2 and 3. For STypes not equal to 0 but otherwise specified in this Standard, PType must = 0, and no message text may be transmitted. For STypes defined in subsidiary standards, the meaning of header bytes 2 and 3 may be specified on a per SType value basis, and these STypes may optionally define message text as long as the PType field is used in a manner consistent with the preceding paragraph.

9.4 *SECS-II Considerations* — The SECS-II Standard (SEMI E5) makes certain references to SECS-I (SEMI E4). This section addresses issues specific to SECS-II when HSMS is used to transport SECS-II messages.

9.4.1 *Reply Matching* — When a Sender sends a Primary Message with W-Bit 1 (Reply Expected), the Sender should expect a Reply message whose header meets the following requirements.

- The SessionID of the Reply must match the SessionID of the Primary Message.
- The Stream of the Reply must match the Stream of the Primary Message.
- The Function of the Reply must be one greater than the Function of the Primary Message, or else the Function of the Reply must be 0 (Function Zero Reply).
- The System Bytes of the Reply must match the System Bytes of the Primary Message.

9.4.1.1 *T3 Reply Timeout* — The T3 reply timeout is a limit on the length of time that the HSMS message protocol is willing to wait for a Reply message.

9.4.1.2 After sending a Primary Message with W-Bit 1 (Reply Expected), the sender must begin a reply timer, initialized to the T3 value. If the sender does not receive the Reply Message before the reply timer expires, then a T3 Timeout Error has occurred. The sender should close the transaction and no longer expect the Reply Message.

9.4.1.3 Each open transaction for which a Reply is expected requires a separate reply timer.



9.4.2 Stream 9 Messages — The SECS-II Standard defines error messages S9F1, S9F3, S9F5, S9F7, S9F9, and S9F11, with message text containing the SECS-II Data Items MHEAD or SHEAD, which are defined to contain a 10-byte SECS-I block header.

9.4.2.1 When using SECS-II with HSMS, MHEAD and SHEAD should contain the ten bytes of the HSMS Message Header.

10 HSMS Documentation

10.1 An HSMS implementation is required to document the following information:

1. Method for setting protocol parameters (see § 10.1).
2. Range allowed and resolution for each parameter.
3. The option used for refusing incoming connection requests if the implementation uses the passive mode for TCP/IP connection establishment.
4. Maximum message size which can be received.
5. Maximum expected size of messages sent.
6. Maximum number of supported concurrent open transactions.

10.2 Parameter Setting — Implementations of HSMS must provide for installation time setting of the following parameters. The range and resolution of all parameters must be at least as shown in the table. All parameters must be stored in such a manner that the settings shall be retained if the power fails or if the system software is reloaded.

Table 10 Parameter Setting

Parameter Name	Value Range	Resolution	Typical Value	Description
T3 Reply Timeout	1–120 seconds	1 second	45 seconds	Reply timeout. Specifies maximum amount of time an entity expecting a reply message shall wait for that reply.
T5 Connect Separation Timeout	1–240 seconds	1 second	10 seconds	Connection Separation Timeout. Specifies the amount of time which must elapse between successive attempts to connect to a given remote entity.
T6 Control Transaction Timeout	1–240 seconds	1 second	5 seconds	Control Transaction Timeout. Specifies the time which a control transaction may remain open before it is considered a communications failure.
T7 NOT SELECTED Timeout	1–240 seconds	1 second	10 seconds	Time which a TCP/IP connection can remain in NOT SELECTED state (i.e., no HSMS activity) before it is considered a communications failure.
T8 Network Intercharacter Timeout	1–120 seconds	1 second	5 seconds	Maximum time between successive bytes of a single HSMS message which may expire before it is considered a communications failure.
Connect Mode	PASSIVE, ACTIVE	-	-	Connect Mode. Specifies the logic this local entity shall use during HSMS connection establishment.
Local Entity IP Address and Port number	determined by TCP/IP conventions	-	-	Required for any entity operating in PASSIVE mode. Determines the address on which the local entity shall listen for incoming connection requests.
Remote Entity IP Address and Port Number	determined by TCP/IP conventions	-	-	Required for any entity operating in ACTIVE mode. Determines the address of the remote entity to which the local entity shall attempt to connect.

#1 Parameter defaults shown above are for small networks (10 nodes or less). Settings may need to be adjusted for larger network configurations.



APPENDIX 1

NOTICE: The material in this Appendix is an official part of SEMI E37 and was approved by full letter ballot procedures.

A1-1 TCP/IP Procedures Using TLI and BSD Socket Interfaces

A1-1.1 Passive Mode Connect Procedure

Table A1-1 Passive Mode Connect Procedure

Intended Action	TLI Construct	BSD Construct	Comment
Obtain a connection endpoint and bind it to a published port.	tep = t_open(...) t_bind(tep,...)	skt = socket(...) bind(skt,...)	BSD refers to a connection endpoint as a ‘socket.’ TLI refers to it as a TEP (transport endpoint).
Permit socket to listen for connections.	...	listen(skt...)	In TLI, the equivalent of BSD listen is not necessary.
Connect procedure: receive incoming connect request and accept it.	t_listen(tep,...) t_accept(tep,...)	accept(skt,...)	
Connect procedure: receive incoming connect request, but reject it.	t_listen(tep,...) t_snndis(tep,...)	...	The BSD API does not support originating a reject of a connect request, as receiving request and accepting it are a single operation.

A1-1.2 Active Mode Connect Procedure

Table A1-2 Active Mode Connect Procedure

Intended Action	TLI Construct	BSD Construct	Comment
Obtain a connection endpoint.	tep = t_open(...) t_bind(tep,...)	skt = socket(...)	TLI requires bind to null address for active entity.
Connect procedure: send connect request and receive accept or reject from passive entity.	t_connect(tep,...) t_rcvconnect(tep,...)	connect(skt,...)	The BSD connect will correctly handle an active reject from the TLI-based remote entity.

A1-1.3 Terminating the Connection

Table A1-3 Terminating the Connection

Intended Action	TLI Construct	BSD Construct	Comment
Release the connection and free connection endpoint.	t_sndrel(tep,...) t_close(tep)	close(skt)	The ‘gracefulness’ of the BSD close is a function of the local implementation.
Disconnect and free the connection endpoint.	t_snndis(tep,...) t_close(tep)	shutdown(skt,2) close(skt)	Shutdown immediately disables further sends and receives if the second arg = 2.



A1-1.4 Sending and Receiving HSMS Messages

Table A1-4 Sending and Receiving HSMS Messages

Intended Action	TLI Construct	BSD Construct	Comment
Send an HSMS message	hdr->Len = length; t_snd(tep, hdr, 14, 0); t_snd(tep, Text, hdr->Len, 0);	hdr->Len = length; write(skt, hdr, 14); write(skt, Text, hdr->Len);	The procedure illustrates a ‘typical’ implementation style in which the length bytes and header are combined into a single 14-byte item sent first, followed by the text. This is not to imply that combining everything is not permitted.
Receive an HSMS message	t_rcv(tep, hdr, 14, ...); t_rcv(tep, Text, hdr->Len, ...);	read(skt, hdr, 14); read(skt, Text, hdr->Len);	As above, but for receiving.

A1-2 HSMS Scenarios

A1-2.1 The following scenarios are provided to illustrate the HSMS procedures as used for a typical complete session. The terminology, procedure names, and message names are further explained in the remainder of this Document. Also note that either entity may initiate the HSMS Select procedure, the Deselect or Separate procedures, and HSMS Data Messages and Transactions. For convenience, the scenarios show the left-hand entity as the initiator of all transactions.

A1-2.2 *Begin HSMS Communication* — This scenario illustrates the TCP/IP connection procedure, an HSMS select procedure, and exchange of data messages. Note that the data message activity for TCP is for illustrative purposes only. In fact, the actual network activity can vary. For example, even if the data messages are sent as separate calls to t_snd (or write) as shown, the TCP/IP implementation may buffer the header and transmit it in a single packet with the text, or the text may be split into multiple packets.

Table A1-5 Begin HSMS Communication

BSD API Calls	TLI API Calls	Network Activity	TLI API Calls	BSD API Calls
Prepare to initiate a connection request. skt = socket(...);	tep = t_open(...); t_bind(tep, ...);		Prepare to receive a connection request. tep = t_open(...); t_bind(tep,...);	skt = socket(...);bind(skt,...);listen(skt,...);
Initiate a connection request and wait for response. connect(skt,...);	t_connect(tep,...); t_rcvconnect(tep,...);	TCP/IP Connect Req Msg(s) TCP/IP Accept Msg(s)	t_listen(tep,...);t_acce pt(tep,...);	accept(skt,...);
Initiate an HSMS Select procedure: send request and receive response. write(skt,hdr,14); read(skt,hdr,14);	t_snd(tep,hdr,14,0); t_rcv(tep,hdr,14,...);	Select.req Message Select.rsp Message	t_rcv(tep,hdr,14,...); t_snd(tep,hdr,14,0);	read(skt,hdr,14); write(skt,hdr,14);
Send an HSMS data message as length bytes and header, followed by Text. hdr->Len = Length; write(skt,hdr,14); write(skt,Text,...);	hdr->Len = Length; t_snd(tep,hdr,14,0); t_snd(tep,Text,...);	HSMS Data Message (hdr) HSMS Data Message (text)	t_rcv(tep,hdr,14,...); t_rcv(tep,Text,...);	read(skt,hdr,14);read(skt,Text,...);



A1-2.3 Ending Communicaiton Using Deselect — This scenario illustrates ending an HSMS Session using the Deselect procedure to end the HSMS session.

Table A1-6 Ending Communication Using Deselect

BSD API Calls	TLI API Calls	Network Activity	TLI API Calls	BSD API Calls
Send the Deselect.req and receive Deselect.rsp.			Receive the Deselect.req and send the Deselect.rsp.	
write(skt,hdr,14); read(skt,hdr,14);	t_snd(tep,hdr,14,0); t_rcv(tep,hdr,14,...);	Deselect.req Message Deselect.rsp Message	t_rcv(tep,hdr,14,...); t_snd(tep,hdr,14,0);	read(skt,hdr,14); write(skt,hdr,14);
Disconnect the TCP/IP Connection.			Respond to Disconnect of connection.	
shutdown(skt,2); close(skt);	t_snddis(tep); t_close(tep);	TCP/IP Disconnect Msg(s)	t_rcvdis(tep,...);	close(skt);

A1-3 HSMS Alternating Mode Connect Procedure

A1-3.1 Some users have particular requirements which prevent them from determining which connect mode (active or passive) a given entity will use at any particular time. In such a case, a Local Entity alternately attempts the active mode and passive mode connect procedure until a connection is successfully established. Note that this requires that local entity provide a published port when in the passive phase. The general logic sequence at the Alternating Local Entity is as follows:

1. Attempt an active connect procedure as described in § 6.3.3 using a timeout value for the t_rcvconnect greater than or equal to the connection separation timeout T5.
2. If the active connect procedure completes successfully, the alternating mode connect procedure completes successfully.
3. If the active connect procedure terminates unsuccessfully, attempt the passive connect procedure as described in § 6.3.2 with the timeout for the t_listen greater than or equal to the connection separation timeout T5.
4. If the passive connect procedure completes successfully, the alternating mode connect procedure completes successfully as described in § 6.3.2.
5. If the passive connect procedure terminates unsuccessfully, the local entity may either return to step 1 to continue the alternating mode procedure or terminate unsuccessfully. The number of times the above sequence of steps are repeated in attempting to form a connection is a local entity-specific issue.

A1-3.2 Alternating Mode Cycle Time — The Alternating Mode Cycle Time is the time between iterations of the Connect Procedure of an Alternating Mode entity. In the above procedure, this corresponds to the duration between the initiation of step 1 and completion of step 5 immediately prior to the reinitiation of step 1. This time is implementation-dependent.

A1-3.2.1 In the case that two entities are both using the Alternating Mode Connect Procedure, it is desirable to ensure that they both have different alternating mode cycle times, to prevent the entities from attempting to connect in lock step: both in active mode, then both in passive mode. Adjusting the Alternating Mode Cycle Time can be readily achieved by adjustment of T5 so that the cycle time is different for the two entities:

A1-3.3 HSMS Connect Combinations — An entity configured as alternating between active and passive mode can connect with either a passive or active mode remote entity. The list below summarizes the combinations possible using the standard with this particular connections strategy.

1. An Entity ‘A’ configured as ACTIVE can connect to an Entity ‘B’ configured as PASSIVE or as ALTERNATING, and Entity A always establishes the Connection.
2. An Entity ‘A’ configured as ALTERNATING can connect to an Entity ‘B’ configured as PASSIVE, and Entity ‘A’ always establishes the Connection.



3. An Entity ‘A’ configured as ALTERNATING can connect to an Entity ‘B’ configured as ALTERNATING, and either end can establish the Connection. In implementations which use multi-threaded connect logic, rather than the sequential logic described in this Document, it may be possible that both ends of the HSMS connection attempt to connect at the same time. In this case, there can be two separate TCP/IP connections established, and it is necessary to establish a convention so that one connection is allowed to remain and the other is terminated.
4. It is not allowed to connect two Entities both configured as PASSIVE, or both configured as ACTIVE.

A1-4 Non-HSMS TCP/IP Protocols

A1-4.1 For typical TCP/IP implementations, HSMS can co-exist with other TCP/IP based protocols on the same IP Address. This can be very useful. For example, a SECS-II message transaction could trigger an application to begin a TCP/IP FTP (File Transfer Protocol) sequence to transfer a large data file.

A1-5 Non-TCP/IP Protocols

A1-5.1 The use of protocols other than TCP/IP on the same network as the HSMS entities is possible but beyond the scope of this Standard. Typically, other protocols could be used, provided they have no impact on TCP/IP or HSMS entities on the network.

A1-6 Multiple LANs

A1-6.1 The HSMS specification considers only a single TCP/IP LAN. Interconnecting multiple LANs is outside the scope of HSMS. However, since TCP/IP implementations typically support such configurations seamlessly through gateways, routers, and similar entities, it may be possible to establish an HSMS Connection across interconnected LAN’s.

A1-7 TCP/IP Physical Layer

A1-7.1 HSMS does not specify the physical layer of IP. Any physical layer supported by TCP/IP can be used. Most commonly, TCP/IP implementations use Ethernet (IEEE 802.3) as the physical layer. However, some TCP/IP implementations use other protocols (e.g., Token Bus, IEEE 802.4 and .5). To ensure interoperability within a given installation, it may be desirable to establish additional local standards for the physical layer.

A1-8 Well-Known TCP/IP Port Numbers

A1-8.1 Some TCP/IP-based protocols specify a particular ‘Well Known’ TCP Port Number, which is published and is not available for other protocols. HSMS does not specify a particular ‘Well Known’ TCP Port Number, but instead requires that it be configurable. The IETF defines ‘Assigned Well Known Port Numbers’ in RFC 1340.

A1-9 Delay Between Disconnect and Re-Connect

A1-9.1 Some TCP/IP systems exhibit problems with applications which terminate a connection and then quickly re-connect, when using identical TCP ports on both ends of the connection. When using such systems, it may be advisable to delay after disconnect before re-connecting. The delay time varies among TCP/IP implementations, but typically can be calculated as twice the ‘maximum segment lifetime’ or MSL. For example, most TCP/IP systems based on BSD (e.g., Sun, AIX) use a MSL of 30 seconds, so a delay of 60 seconds would be appropriate. If rapid connects are required, your application should use a different Port, if allowed by the Connect Mode you are using. In some TCP/IP systems, this problem does not occur.

A1-10 User-Defined Message Types

A1-10.1 It is recognized that equipment suppliers may find it desirable to develop additional features not found in the base level HSMS or any defined subsidiary standards. This will be the case during the testing and development of any proposed new subsidiary standard. User-defined extensions through new message types are permissible as long as they are confined to intra-vendor communication interfaces: any inter-vendor communications interface which requires the use of such extensions is considered to be noncompliant with the HSMS standard.

A1-10.2 If a supplier does deem it necessary to extend or otherwise go outside the standard, the use of ‘reserved, not used,’ values of PType and SType may simplify their implementation by permitting the reuse of the HSMS implementation rather than the implementation and use of a completely separate parallel standard. By remaining



within the ‘reserved, not used,’ ranges for SType and PType, the implementor can be assured that future subsidiary standards which define new values for SType and/or PType will not conflict with user-defined extensions.

A1-11 Comparison of SECS-I and HSMS

A1-11.1 The following table compares major features of SECS-I and HSMS.

Table A1-7 Comparison of SECS-I and HSMS

Feature	SECS-I	HSMS
Communications Protocol Base	RS-232	TCP/IP
Physical Layer	25-pin connector and 4-wire serial cable	Physical layer not defined. HSMS allows any TCP/IP supported physical medium. Typical example is Ethernet (IEEE 802.3) and thin coax (10-BASE-2).
Communications Speed	Typically about 1000 bytes/second (assuming 9600 baud).	Typically 10 MBits/second (assuming typical Ethernet).
Connections	One physical RS-232 cable per SECS-I connection.	One physical network cable can support many HSMS Connections.
Message Format	Message text is SECS-II Data Items. Transmits a SECS-II message as a series of transmittal blocks each approximately 256 bytes in size. Each block has a one-byte block length, a ten-byte Block Header, text, and a two-byte Checksum.	Message Text is SECS-II Data Items. Transmits a SECS-II message as a TCP/IP byte stream. The message has a four-byte Message Length, a ten-byte Message Header, and text. The TCP/IP layer may impose blocking limits which depend on the physical layer used, but this blocking is transparent to the TCP/IP API and is outside the scope of HSMS.
Header	Ten-byte header on each block of a message. Header bytes 4–5 contains E-Bit and Block Number.	One ten-byte Header for the entire message. Header bytes 4–5 contain PType and SType. Header bytes 2–3 are W-Bit, Stream, and Function when SType = 0 (Data Message). For SType not equal to 0 (Control Message), bytes 2–3 have other uses. No R-Bit.
Maximum message size	Limited to approximately 7.9 million bytes (32767 blocks times 244 text bytes per block).	Message size limited by 4-byte message length (approximately 4 GBytes). Local implementation of TCP/IP and HSMS may further limit this in practice.
Protocol Parameters (Common)	T3 Reply Timeout Device ID.	T3 Reply Timeout Session ID (analogous to Device ID).
Protocol Parameters (SECS-I only)	Baud Rate T1 Inter-Character Timeout T2 Block Protocol Timeout T4 Inter-Block Timeout RTY Retry Count Host/Equipment	Not used in HSMS. Corresponding issues addressed by TCP/IP layers.
Protocol Parameters (HSMS Only)	Not needed by SECS-I.	IP Address and Port of Passive Entity. T5 Connect Separation Timeout. T6 Control Transaction Timeout. T7 NOT SELECTED Timeout. T8 Network Intercharacter Timeout.



NOTICE: SEMI makes no warranties or representations as to the suitability of the Standards and Safety Guidelines set forth herein for any particular application. The determination of the suitability of the Standard or Safety Guideline 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. Standards and Safety Guidelines are subject to change without notice.

By publication of this Standard or Safety Guideline, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this Standard or Safety Guideline. Users of this Standard or Safety Guideline 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.





SEMI E37.1-0702 (Reapproved 0413)

HIGH-SPEED SECS MESSAGE SERVICE SINGLE SELECTED-SESSION MODE (HSMS-SS or HSMS-SSS)

This Standard was technically approved by the Information & Control Global Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on December 20, 2012. Available at www.semiviews.org and www.semi.org in April 2013; originally published in 1995; previously published March 2008.

NOTICE: This Document was reapproved with minor editorial changes.

1 Purpose

- 1.1 High-Speed SECS Message Services Single-Session Mode (HSMS-SS) provides a means for independent manufacturers to produce implementations which can be connected without requiring specific knowledge of one another.
- 1.2 HSMS-SS is intended as an alternative to SEMI E4 (SECS-I) for applications where higher speed communication is needed.
- 1.3 HSMS-SS is intended as an alternative to SEMI E13 (SECS Message Services) for applications where TCP/IP is preferred over OSI as a communications basis.

2 Scope

- 2.1 HSMS-SS is a subsidiary standard to High-Speed SECS Message Services (HSMS) Generic Services.

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

3 Reference Standards and Documents

3.1 SEMI Standards and Safety Guidelines

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

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

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

4 Terminology

4.1 Definitions

4.1.1 *device ID* — a 15-bit field in the message header used to identify a sub-entity within the equipment.

4.2 In addition, all definitions for HSMS Generic Services apply.

4.3 Note that the terms HSMS and HSMS generic services both refer to the HSMS Generic Services standard definition [SEMI E37].

5 HSMS-SS Overview and State Machine

5.1 This definition defines the HSMS-SS-specific use of HSMS Generic Services suitable for applications requiring a simple SECS-I replacement. The purpose of this Standard is to explicitly limit the capabilities of the HSMS Generic Services to the minimum necessary for this type of application. Specifically, HSMS imposes the following limitations:

1. HSMS-SS eliminates the use of a number of HSMS procedures. Deselect is not to be used to end HSMS-SS communications (use Separate instead), and the Reject procedure is optional.



2. HSMS-SS limits certain other procedures such as Select to simplify operation for the specific case of SECS-I replacement.

5.2 The remainder of this Document describes these limitations in more detail.

5.3 *HSMS-SS State Machine* — The HSMS-SS behavior and state machine differ from that specified in the HSMS Generic Services in the following ways:

1. The SelectionCounter defined in HSMS Generic Services is not required.
2. Various transitions are defined differently as illustrated in the HSMS-SS state machine illustrated below.

5.4 The HSMS-SS state machine is illustrated in the diagram below.

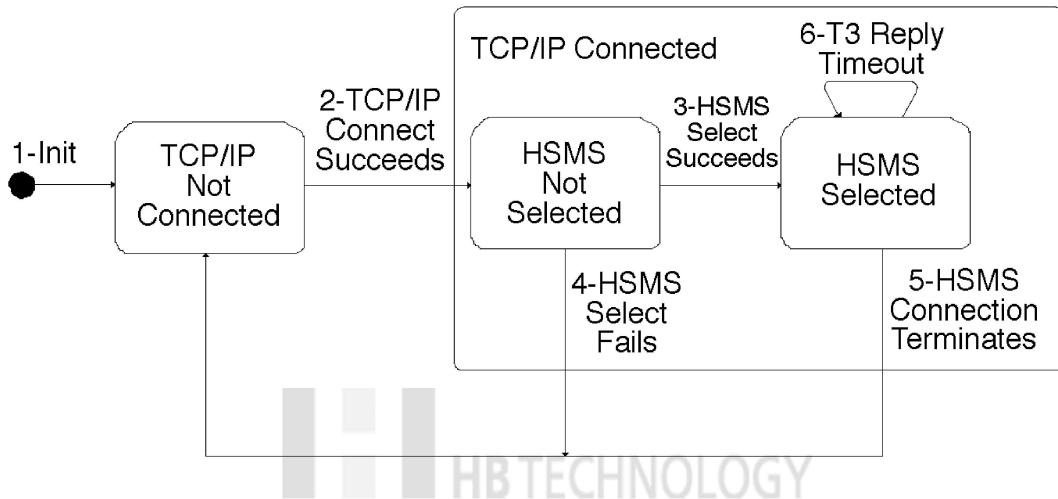


Figure 1

5.5 State Transition Table for Passive Mode Connect

Table 1 HSMS-SS Passive Mode Connect State Transitions

#	Old State	New State	Trigger	Actions
1	—	TCP/IP NOT CONNECTED	Initialization.	
2	TCP/IP NOT CONNECTED	HSMS NOT SELECTED	TCP/IP Connect Succeeds: 1. TCP/IP “accept” succeeds.	Start T7 timeout.
3	HSMS NOT SELECTED	HSMS SELECTED	HSMS Select Succeeds: 1. Receive Select.req and decide to allow it.	1. Cancel T7 timeout; and 2. Send Select.rsp with zero SelectStatus.



#	<i>Old State</i>	<i>New State</i>	<i>Trigger</i>	<i>Actions</i>
4	HSMS NOT SELECTED	TCP/IP NOT CONNECTED	HSMS Select Fails: 1. T7 Timeout waiting for Select.req; or 2. Receive Select.req and decide to reject it and send Select.rsp with non-zero SelectStatus; or 3. Receive any HSMS message other than Select.req; or 4. Receive HSMS message length not equal to 10; or 5. Receive bad HSMS message header; or 6. T8 timeout waiting for TCP/IP; or 7. Other unrecoverable TCP/IP Error (entity-specific).	1. Close TCP/IP connection.
5	HSMS SELECTED	TCP/IP NOT CONNECTED	HSMS Connection Terminates: 1. Decide to terminate and send Separate.req; or 2. Receive Separate.req; or 3. T6 timeout waiting for Linktest.rsp; or 4. Receive HSMS message <10; or 5. Receive HSMS message length > maximum supported by entity; or 6. Receive bad HSMS message header; or 7. T8 timeout waiting for TCP/IP; or 8. Other uncorrectable TCP/IP Error (entity-specific).	1. Close TCP/IP connection.
6	HSMS SELECTED	HSMS SELECTED	T3 Timeout waiting for Data Reply Message.	1. Cancel the Data Transaction as appropriate (entity-specific) but do not terminate the TCP/IP connection; and 2. If entity is EQUIPMENT send SECS-II S9F9.

5.6 State Transition Table for Active Mode Connect

Table 2 HSMS-SS Active Mode Connect State Transitions

#	<i>Old State</i>	<i>New State</i>	<i>Trigger</i>	<i>Actions</i>
1	—	TCP/IP NOT CONNECTED	Initialization	
2	TCP/IP NOT CONNECTED	HSMS NOT SELECTED	TCP/IP Connect Succeeds: 1. Decide to connect.	1. TCP/IP Connect; and 2. Send Select.req; and 3. Start T6 timeout.
3	HSMS NOT SELECTED	HSMS SELECTED	HSMS Select Succeeds: 1. Receive Select.rsp with zero SelectStatus.	1. Cancel T6 timeout.



#	Old State	New State	Trigger	Actions
4	HSMS NOT SELECTED	TCP/IP NOT CONNECTED	HSMS Select Fails: 1. T6 Timeout waiting for Select.rsp; or 2. Receive Select.rsp with non-zero Select.Status; or 3. Receive any HSMS message other than Select.rsp; or 4. Receive HSMS message length not equal to 10; or 5. Receive bad HSMS message header; or 6. T8 timeout waiting for TCP/IP; or 7. Other unrecoverable TCP/IP Error (entity-specific).	1. Close TCP/IP connection; and 2. Start T5 Timeout.
5	HSMS SELECTED	TCP/IP NOT CONNECTED	HSMS Connection Terminates: 1. Decide to terminate and send Separate.req; or 2. Receive Separate.req; or 3. T6 timeout waiting for Linktest.rsp; or 4. Receive HSMS message length < 10; or 5. Receive HSMS message length > maximum supported by entity; or 6. Receive bad HSMS message header; or 7. T8 timeout waiting for TCP/IP; or 8. Other uncorrectable TCP/IP Error (entity-specific).	1. Close TCP/IP connection.
6	HSMS SELECTED	HSMS SELECTED	T3 Timeout waiting for Data Reply Message.	1. Cancel the Data Transaction as appropriate (entity-specific) but do not terminate the TCP/IP connection; and 2. If entity is EQUIPMENT, send SECS-II S9F9.

Table 3 When HSMS Transactions are Allowed

HSMS Transition	Allowed in State(s)	Who Initiates Transaction?
Select	HSMS Not Selected	Active Entity
Link Test	HSMS Selected	Either Entity
Data	HSMS Selected	Either Entity
Separate	HSMS Selected	Either Entity

6 HSMS-SS Use of TCP/IP

6.1 As defined in HSMS.

7 HSMS-SS Procedures

7.1 *Select Procedure* — The Select Procedure shall only be initiated by the entity establishing the TCP/IP connection in active mode. The Passive Mode Entity shall not initiate the Select Procedure.

7.1.1 The Select Procedure is only permitted in the NOT SELECTED state. It uses a SessionID value of 0xFFFF and implies that all device IDs are available for communication. Immediately following any Select Procedure which fails to complete successfully with a zero Select Status, each Entity must close the TCP/IP connection and transit to the NOT CONNECTED state.



7.2 Data Procedure — The Data Procedure is as defined in HSMS Generic Services. Note that any SessionID value that corresponds with a DeviceID supported by the Local Entity is valid as long as the Local Entity is in the SELECTED state.

7.3 Deselect Procedure — Deselect shall not be used in an HSMS-SS implementation. Communication is ended using the Separate Procedure.

7.4 Linktest Procedure — As defined by HSMS. Under HSMS-SS, the use of Linktest is strictly limited to the SELECTED state.

7.5 Reject Procedure — The Reject Procedure is optional in HSMS communications. Note, however, that any situation which would require the use of the Reject as described in HSMS Generic Services shall be treated as a communications failure in implementations not supporting reject. Specifically, the TCP/IP connection is immediately closed.

7.6 Separate Procedure — Separate shall always use SessionID 0xFFFF (binary, all ones). In HSMS-SS, the Separate.req is valid only in the TCP/IP CONNECTED state and its substates. After either initiating or receiving a Separate.req message, the entity shall immediately close the TCP/IP connection and transit to the TCP/IP NOT CONNECTED state.

7.7 Communications Failures — As defined by HSMS. Note that, in addition to the communications failures defined under HSMS, any violation of the restrictions defined in prior sections of this Document are also to be treated as communication failures.

8 HSMS-SS Message Format

8.1 Session ID — In HSMS-SS Data Messages, the high-order bit of Session ID is zero, and the low-order 15 bits contain Device ID, a 15-bit unsigned integer value, which occupies the low-order 7 bits (bits 6-0) of byte 0 and all of byte 1 of the header. Device ID is a property of the equipment, and can be viewed as a logical identifier associated with a physical device or sub-entity within the equipment. The precise meaning of ‘device’ or ‘sub-entity’ is equipment-defined. A unit of equipment must have at least one Device ID. Equipment which contains several devices may define a unique Device ID for each device.

8.1.1 In HSMS-SS Control Messages, Session ID will always assume the special value 0xFFFF (all one bits).

8.2 PType — All HSMS-SS messages are PType 0 (SECS-II encoded) as defined in HSMS.

8.3 STType — Only HSMS-defined STypes are permitted in HSMS-SS. User-defined STType messages are not permitted.

9 Special Considerations

9.1 Multiblock Messages — For each SECS-II message, the SECS-II standard defines whether that message should be transmitted in SECS-I as a single-block message or as a multiblock message.

9.2 This distinction becomes unimportant with HSMS, which transmits all messages in the same fashion. However, to be compatible with older SECS-I applications, when an HSMS application sends a SECS-II message defined as single block, the HSMS Message Length should not exceed 254 bytes (10 byte header plus 244 text bytes).

10 HSMS-SS Documentation

10.1 An HSMS-SS implementation is required to document the following information in addition to the information required by HSMS.

1. The number of deviceIDs supported and their specific values.
2. Whether or not the implementation supports the normal or the restricted procedure for terminating communications.
3. The setting of the host vs. equipment parameter.

10.2 Host vs. Equipment — Many applications using SECS-II will need to designate one end of the communication link as ‘Equipment’ and the other end as ‘Host.’ HSMS-SS itself does not require configuring of ‘Host’ and

Licensed by SEMI to hlji@hbtechnology.co.kr (252095)
Downloaded: 1/9/2019
Single-user license only, copying and networking prohibited.

Status:CURRENT



'Equipment,' but this parameter may be included in configuration where needed. HSMS can also be used in applications where the distinction between Host and Equipment is not used.





RELATED INFORMATION 1 APPLICATION NOTES

NOTICE: This Related Information is not an official part of SEMI E37.1 and was derived from the work of the Information & Control Global Technical Committee. This Related Information was approved for publication by full letter ballot procedures.

R1-1 Introduction

R1-1.1 An entity may have more than one session. If a unit of equipment can be divided into two or more logical sub-equipments, such as process chambers, or process resources, each of these may have separated session ID. If two or more physical sub-equipments are controlled by an equipment controller or communicated to through a TCP/IP network device, each sub-equipment may have a separate session ID. Such a session can be established once HSMS is selected. Each session ID corresponds to a device ID.

R1-1.2 If a unit of equipment has more than one sub-equipment or process resource (e.g., process modules) but the sub-equipment share a common resource (e.g., transfer subsystem), it is not recommended that each sub-equipment have an independent session to communicate with host. Even if the host requests an action to the shared subsystem with a session identifier specific for a sub-equipment, the shared subsystem may not always serve for the sub-equipment. Since the host expects the shared subsystem to perform the service for the sub-equipment, an error will occur if it is not possible to perform the service for the designated sub-equipment. See following figure.

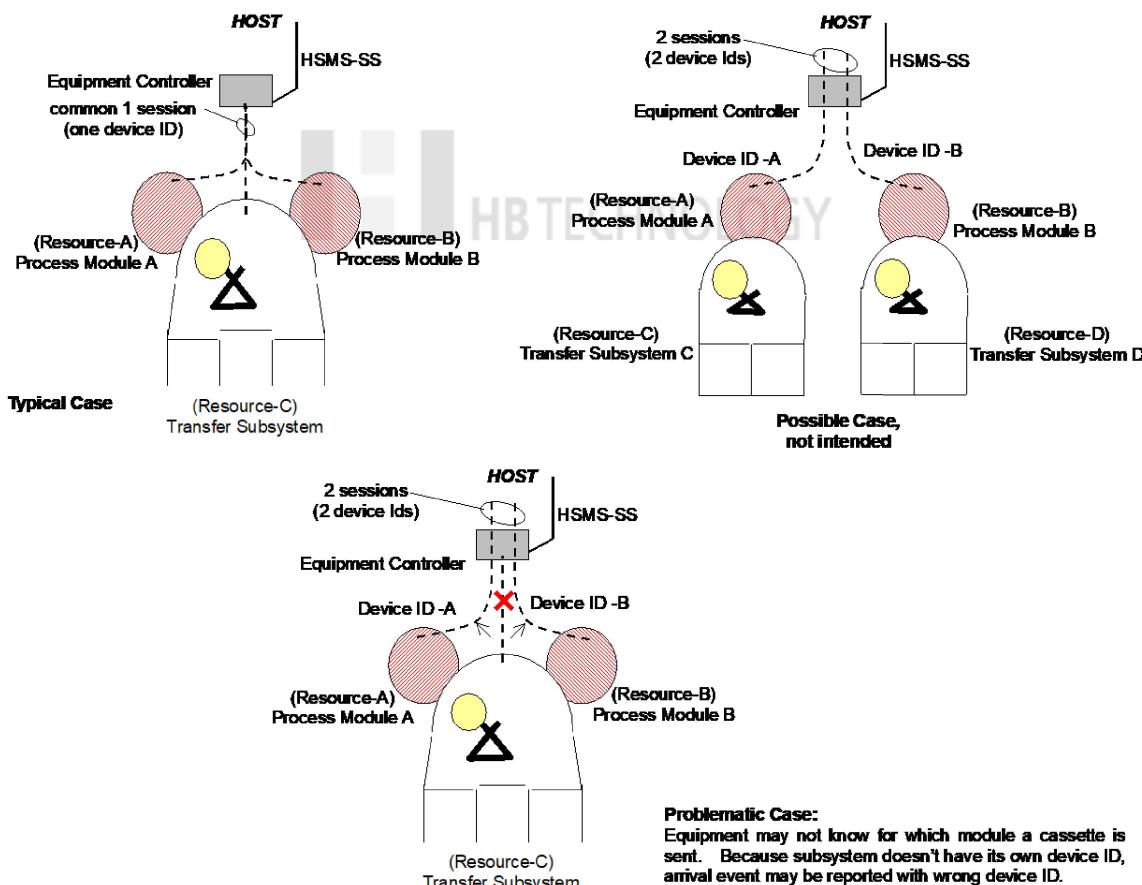


Figure R1-1



R1-2 Multiple HSMS Connections

R1-2.1 Typically, an Equipment will accept only one Host Connection.

R1-2.2 A Host may connect to several units of Equipment, so the Host may have several simultaneously active Connections (each to one Equipment).

R1-2.3 A Cell Controller (or similar entity) might have one Connection by which the Cell Controller appears as 'Equipment' to the Factory Host Computer, as well as several Connections by which the Cell Controller appears as 'Host' to Equipment.

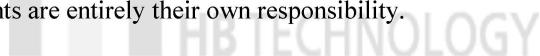
R1-3 Equipment Support for Multiple Hosts

R1-3.1 HSMS requires Equipment to accept at least one active Connection, and does not require the equipment to support access by multiple concurrent Hosts. That is, if the Equipment has already accepted a Host Connection, but a Host (the same or a different Host) attempts a second Connection, the Equipment will immediately terminate that second Connection attempt.

R1-3.2 For specialized applications, an equipment could accept more than one Host Connection. Coordination of activity by multiple hosts is equipment-defined.

NOTICE: SEMI makes no warranties or representations as to the suitability of the Standards and Safety Guidelines set forth herein for any particular application. The determination of the suitability of the Standard or Safety Guideline 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. Standards and Safety Guidelines are subject to change without notice.

By publication of this Standard or Safety Guideline, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this Standard or Safety Guideline. Users of this Standard or Safety Guideline 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.





SEMI E37.2-95 (Withdrawn 1109)

HIGH-SPEED SECS MESSAGE SERVICES GENERAL SESSION

(HSMS-GS)

This Standard was technically approved by the Information & Control Global Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on September 4, 2009. Available at www.semi.org in November 2009; originally published in 1995; previously published March 2003.

NOTICE: This Document was balloted and approved for withdrawal in 2009.

1 Purpose

1.1 HSMS-GS is intended to support the needs of complex systems containing multiple independently accessible subsystems such as cluster tools or track systems. Specifically, procedures are defined to permit access to any individual subsystem or set of subsystems within any complex system.

2 Scope

2.1 High-Speed SECS Message Services General Session (HSMS-GS) is a subsidiary standard to High-Speed SECS Message Services (HSMS) Generic Services.

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

3 Referenced Standards and Documents

3.1 SEMI Standards and Safety Guidelines

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

SEMI E5 — Specification for SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E37 — Specification for High-Speed SECS Message Services (HSMS) Generic Services

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

4 Terminology

4.1 Definitions

4.1.1 *Selected Entity List* — a list of session entities currently selected for communication on a given TCP/IP connection.

4.1.2 *Selection Count* — the number of sessions opened by an HSMS Select procedure and not yet ended by an HSMS Deselect or Separate procedure.

4.1.3 *Session Entity* — an individually selectable entity within an HSMS-GS system.

4.1.4 *Session Entity ID* — a 16-bit identifier for a Session-Entity.

4.1.5 *SessionEntityList* — a list of all available session entities within an HSMS-GS system and associated with a particular IP address and port number.

4.2 In addition, all definitions for HSMS Generic Services apply.

4.3 Note that the terms HSMS and HSMS Generic Services both refer to the HSMS Generic Services standard definition (SEMI E37).

5 HSMS-GS Overview and State Machine

5.1 HSMS-GS provides a set of definitions which permit the individual sub-entities (e.g., subsystems) of complex entities (e.g., systems) to be separately accessible during HSMS procedures. HSMS-GS defines no new procedures or message types beyond HSMS Generic Services to provide these services. It does, however, require extensions to the



HSMS State machine, in the form of additional state transition definitions and additional state information, which are used by the extended state machine which must be maintained by an HSMS-GS implementation to support the extended state machine. The additional information consists of the following:

1. The Session Entity List
2. The Selected Entity List
3. The Selection Count

5.2 The Session Entity List consists of the set of all Session Entities having individual accessibility within the HSMS-GS entity. The scope of this list is normally the entire HSMS-GS entity. HSMS-GS, however, does not require this scope: the supplier may provide access to HSMS-GS entity through more than one well known port and provide a different Session Entity List for each.

5.3 A Session Entity is any individually addressable sub-entity within the HSMS-GS entity: for example, a Session Entity may be an individual sub-device in a track system or cluster tool, or may be an individual service provider, such as a data server, within an entity. HSMS-GS only provides the conventions for identifying Session Entities. It places no restrictions on the individual supplier as to what constitutes a SessionEntity: the supplier must determine what is the most appropriate for the particular implementation.

5.4 The Selected Entity List is the list of Session Entities actually selected for access on a given TCP/IP connection. When the TCP/IP connection is established (CONNECTED state entered), an empty Selected Entity List is created. Each time, the Select Procedure is used to select a Session Entity, its Session Entity ID is added to the Selected Entity List created for that TCP/IP connection. Each time the Session Entity is deselected via the Deselect or Separate procedures, its Session Entity ID is removed from the Selected Entity List. At any given time on any given TCP/IP connection endpoint, HSMS Data Messages will only be accepted by an entity if the SessionID of the Data Message is equal to any Session Entity ID in the Selected Entity List.

5.5 The Selection Count is simply the number of Session Entity IDs in the Selected Entity List. Its value affects the behavior of the state machine: the transition to NOT SELECTED from SELECT can only take place when the SELECTION COUNT is zero.

5.6 Note that the above lists and count are defined for the purpose of explaining state machine operation only. There is no requirement that any of the above lists and count be explicitly implemented.

5.7 *HSMS-GS State Machine* — The HSMS-GS state machine is the same as the HSMS Generic Services state machine with the addition of the two new state transitions (#6 and #7) and the use of the Selection Count in the other transitions as described in the state transition table below.

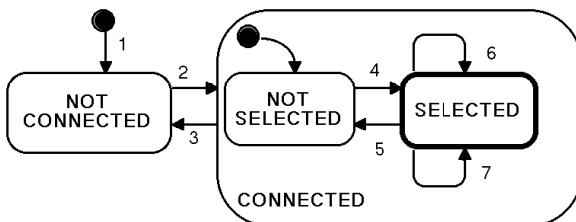


Figure 1

5.8 *State Descriptions* — The state descriptions are the same as HSMS Generic Services.

5.9 *State Transition Table* — The state transition table is almost identical with the state transition table for HSMS Generic Services. For convenience, however, the entire table is reproduced and extended here, not just those areas which differ from it.

**Table 1 State Transition Table**

#	<i>Current State</i>	<i>Trigger</i>	<i>New State</i>	<i>Actions</i>	<i>Comment</i>
1	...	Local entity specific preparation for TCP/IP communication.	NOT CONNECTED	Local entity-specific.	Action depends on connection procedure to be used: active or passive.
2	NOT CONNECTED	A TCP/IP connection is established for HSMS communication.	CONNECTED - NOT SELECTED	Set Selection Count = 0 and create empty Selected Entity List.	none
3	CONNECTED	Breaking of TCP Connection.	NOT CONNECTED	Local entity-specific.	none
4	NOT SELECTED	Successful completion of HSMS Select Procedure.	SELECTED	Set Selection Count = 1 and add selected Session Entity to Selected Entity List.	none
5	SELECTED	Deselect or Separate procedure resulting in Selection Count = 0.	NOT SELECTED	Local entity-specific.	See transition 7 below.
6	SELECTED	Successful completion of HSMS Select Procedure when Selection Count > 0.	SELECTED	Increment Selection Count and add selected Session Entity to Selected Entity List.	none
7	SELECTED	Successful completion of HSMS Deselect or Separate when Selection Count > 1.	SELECTED	Decrement Selection Count and remove selected Session Entity from Selected Entity List.	If this transition results in Selection Count = 0, immediately trigger state transition 5.

6 HSMS-GS Use of TCP/IP

6.1 There is no additional HSMS-GS specification for the use of TCP/IP beyond the above mentioned creation of the empty “Selected Entity List” upon entry into the CONNECTED state, NOT SELECTED substate.

7 HSMS-GS Specific Procedures

7.1 HSMS-GS provides further specification of the following procedures.

7.2 *Select Procedure* — The select procedure is permitted in both the NOT SELECTED state and the SELECTED state. The procedure for both the initiator and the responding entity is the same as HSMS Generic Services, with the following additional conditions:

1. If the responding entity contains no Session Entity in its Session Entity List whose ID matches the SessionID in the Select.req, a Select Status of No Such Entity is used in the Select.rsp.
2. If the responding entity is unable to provide access to the selected entity because it is usable on only a single TCP/IP connection at any one time and it is already in use on a different TCP/IP connection, a response code of Entity In Use is used in the Sel-ect.rsp.
3. If the responding entity finds the SessionID already in its Selected Entity List, a response code of Entity Selected is used in the Select.rsp.

7.2.1 The Select Status values referenced above are all defined in § 8.

7.2.2 If none of the above is true, the Select completes successfully, and a Select Status of 0 is provided in the Select.rsp. Both entities will add the SessionID from the Select.req to the Selected Entity List.

7.3 *Data Procedure* — The Data Procedure is as defined in HSMS Generic Services. Note that the SessionID of any Data Message must match a SessionID in the SelectedEntityList. If a Data Message is received with a SessionID other than one from the Selected Entity List, a Reject.req message is sent in response by the receiving entity. The reason code will be Entity Not Selected.



7.4 Deselect Procedure — The Deselect procedure is restricted by the following conditions:

1. The SessionID must be in the Selected Entity List.
2. The corresponding SessionEntity is in a state which permits Deselect. This decision is local entity specific and not subject to the HSMS-GS.

7.4.1 If both of the above are true, then the Deselect can proceed. Assuming that the Deselect completes successfully, the SessionID is removed from the Selected Entity List, and the Selection Count is decremented. The transition to the NOT SELECTED state transpires only if the resulting Selection Count is equal to zero (i.e., an empty Selected Entity List).

7.5 Linktest Procedure — As defined by HSMS.

7.6 Reject Procedure — As defined by HSMS. Note, in particular, the use of Reject in response to certain data messages (above).

7.7 Separate Procedure — The Separate procedures and state transitions are subject to the same restrictions and conditions as the Deselect.

7.8 Communications Failures — As defined by HSMS. Note that any abrupt termination has the effect of deselecting all entities in the Selected Entity List.

8 HSMS-GS Message Format Issues

8.1 Session ID — In HSMS-GS Select, Data, Deselect, Reject, and Separate messages, the SessionID will equal the Session Entity ID of the target Session Entity which must equal the Session ID of a Session Entity contained in the Session Entity List. In the Linktest, it is 0xFFFF, as in HSMS.

8.2 PType — HSMS-GS messages are generally PType = 0 (SECS-II-encoded) as defined in HSMS. Although other PTYPES are permitted, specific application domains may restrict the use of HSMS-GS to Ptype = 0.

8.3 STYPE — Only HSMS-defined STypes are permitted in HSMS-GS.

8.4 Select/Deselect Status — The following additional enumeration applies to the Select/Deselect status in HSMS-GS:

Table 2 SelectStatus

<i>Value</i>	<i>Description</i>
4	No Such Entity — Session ID does not correspond to any Session Entity ID available at this connection.
5	Entity In Use (by another connection) — Session Entity corresponding to Session ID is not sharable connections and is already selected by another connection.
6	Entity Selected (by current connection) — Session entity corresponding to Session ID is already selected on current connection.

9 Special Considerations

9.1 There are no special considerations above and beyond those described in HSMS Generic Services.

10 HSMS-GS Documentation

10.1 An HSMS-GS implementation is required to document the following information in addition to the information required by HSMS.

1. *The Session Entity List* — Specifically the number of Session Entities available for HSMS-GS communication and the value of their particular Session Entity ID's.



RELATED INFORMATION 1

APPLICATION NOTES

NOTICE: This Related Information is not an official part of SEMI E37.2 and was derived from the work of the Information & Control Global Technical Committee. This Related Information was approved for publication by full letter ballot procedures.

R1-1 Supporting Both HSMS-GS and HSMS-SS Simultaneously

R1-1.1 In certain applications, the equipment manufacturer may be faced with a requirement of providing both HSMS-GS and HSMS-SS communications interfaces. For example, a cluster tool may use HSMS-GS as the intra-tool communications and HSMS-SS as a GEM interface to the factory. However, implementing a communications facility that is simultaneously HSMS-SS- and HSMS-GS- compliant is straightforward.

R1-1.2 Assuming that one has implemented an HSMS-GS, a simple test can be added to the first Select.req received by the equipment. If it is a Select for any particular SessionID, then operate as HSMS-GS. If its value is a -1, then copy the contents of the Session Entity List corresponding TCP/IP IP address and port number into the Selected Entity List. The Selection Counter would be set to a special value, such as -1, to indicate selection in this manner. The Separate would provide the reverse function. If the passive entity is implemented as an HSMS-SS node and the active entity is an implementation that supports both modes of operation, it must be explicitly configured to initiate the Select with a Session ID of -1 and must have a Session Entity List containing the Device IDs of all the available sub-devices within the passive mode entity.

R1-1.3 In the local API for the case where the equipment must originate the select, a configuration parameter for the equipment could indicate which mode to use for a particular remote target.

NOTICE: SEMI makes no warranties or representations as to the suitability of the Standards and Safety Guidelines set forth herein for any particular application. The determination of the suitability of the Standard or Safety Guideline 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. Standards and Safety Guidelines are subject to change without notice.

By publication of this Standard or Safety Guideline, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this Standard or Safety Guideline. Users of this Standard or Safety Guideline 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.