

## SCSI on ST, Version received on 4/12/99

This document mimics the structure of the SCSI on Fibre Channel (FCP) document of the SCSI-3 specification.

### References:

- 1) ST standard
- 2) SCSI-3 standards

The ST Protocol for SCSI standard is part of the SCSI-3 family of standards developed by X3T10 to facilitate the use of the SCSI command sets for many different types of devices across many different types of physical interconnects [not really, but it probably should be...]. The master architectural document of the family of standards is ANSI X3.270-199X, Information Technology - SCSI-3 Architecture Model (SAM). The SAM document contains a guide to other documents in the SCSI-3

Clause 1 describes how SCSI operations are structurally mapped on to ST operations in the SCSI on ST protocol.

Clause 2 defines specific message formats for the SCSI on ST protocol.

Clause 3 defines error recovery procedures.

### 1 General

=====

#### 1.1 Structure and Concepts

=====

Scheduled Transfer (ST) is a data transfer protocol which may be implemented on a wide variety of Lower Layer Protocols (LLP), which supports a connection-oriented data transfer model with multiple outstanding data transfers per connection.

Three kinds of functional management are defined by this SCSI on ST specification:

- 1) Connection management
- 2) Device management
- 3) Task management

For the purposes of describing this protocol, we define the term Transaction which consists of one or more distinct ST Sequences (e.g. Read Sequence, Write Sequence, PMR Put Sequence), which are logically associated to implement a complete SCSI function from the list above.

The correspondence between SCSI functions, elements of the SCSI on ST protocol, and elements of the ST protocol on which SCSI on ST is built are:

SCSI	SCSI on ST	ST
Command Service Request	Request_To_Send or Request_To_Receive or Request_Zero_Length	Request_To_Send operation Request_To_Receive operation NOP operation
Data delivery request	Clear_To_Send operation	Clear_To_Send operation
Data delivery action	Data operation (STU)	Data operation
Command service response	Status Put	Put operation
I/O Operation	Transaction	Read sequence and Put sequence or Write sequence and Put sequence or NOP operation and Put sequence
Task Management Operation	Transaction or Request_Zero_Length or End sequence	NOP operation and Put operation NOP operation End Sequence

### 1.2 Common ST Operation Characteristics

=====

All ST control operations used in the SCSI on ST protocol shall have the ST Interrupt flag bit set.

The non-final ST Data operations of all data sequences (regular data, or Status Put), shall have the ST Interrupt and Last flag bits cleared, and the ST Silent flag bit set.

The final ST Data operation of all data sequences shall have the ST Interrupt and Last flag bits set and the ST Silent flag bit cleared.

All ST Data operations for a SCSI on ST connection shall be sent on the ST Data Channel specified in the connection setup protocol. ST Data Channel flag bits shall be appropriately set to reflect this.

The SCSI on ST protocol does not use the operation pairs for reliable data movement as described in section 10.2 of the ST protocol specification. Instead, entire SCSI on ST Transactions are retried as described in section 3.2, Operation Timeouts.

### 1.3 Connection management

=====

A SCSI on ST connection is an extension of the basic ST Virtual Connection.

### 1.3.1 Connection Setup

=====

A SCSI on ST connection is requested by sending an ST Request\_Connect operation to the well known port for SCSI on ST (TBD), with the Out\_Of\_Order (O) flag bit cleared, and the SCSI on ST connection parameters in the optional payload. SCSI on ST requires that out of order operation not be requested in order to support the Clear\_To\_Send operation accounting protocol described in section 1.4.1, Underrun Residual Handling.

If the connection request is acceptable to the SCSI Target, it responds with an ST Connection\_Answer operation, again with the Out\_Of\_Order (O) flag bit cleared, and with its SCSI on ST connection parameters in the optional payload.

If the SCSI on ST connection parameters are unacceptable to the connection responder, the responder shall reject the Virtual Connection attempt by setting the ST Reject Flag bit, with an appropriate reason code, as specified in section 2.1.6

If the SCSI on ST connection parameters are unacceptable to the connection Initiator, the Initiator shall tear down the Virtual Connection with the standard ST Virtual Connection teardown sequence.

If the SCSI on ST connection parameters indicate that the connection Initiator can function as a SCSI Initiator, and the connection responder intends to function as a SCSI Target, the connection responder shall request the connection Initiator to expose a Persistent Memory Region, called the Status Persistent Memory Region, of a size specified in the Request\_Connect optional payload, for the ST Data Channel specified in the Request\_Connect optional payload, with an ST Request\_Memory\_Region operation. The connection Initiator shall expose the Status Persistent Memory Region and respond with an appropriate Memory\_Region\_Available operation. The initial Offset of the Status Persistent Memory Region shall be a multiple of 512 bytes, the Status Persistent Memory Region segment size.

When the Connection\_Answer operation is received by the connection Initiator, if the connection parameters specify that the connection responder can function as a SCSI Initiator and the connection Initiator intends to function as a SCSI Target, the connection Initiator shall request that the connection responder expose a Status Persistent Memory Region, as described above.

A SCSI on ST Virtual Connection endpoint shall not initiate any SCSI operations until the Status Persistent Memory Region has been requested, exposed and the ST Memory\_Region\_Available operation sent. A SCSI on ST Initiator shall keep a timer awaiting the ST

Request\_Memory\_Region operation, and if none arrives within the duration of the timer, tear down the ST Virtual Connection with the standard ST Virtual Connection teardown sequence and retry the SCSI on ST connection sequence.

A SCSI on ST Target shall respond to any Transaction initiating operation (Request\_To\_Send, Request\_To\_Receive or Request\_Zero\_Length) with a Request\_Answer with a reason code of Status Persistent Memory Region Not Established (TBD) until the ST Memory\_Region\_Available operation is received.

If the initial offset returned in the ST Memory\_Region\_Available operation for the SCSI on ST Status Persistent Memory Region is not a multiple of the the Status Segment Size, 512 bytes, a SCSI on ST Target shall respond to any data transfer request with a Request\_Answer, with a reason code of Status Persistent Memory Region Unaligned (TBD).

If a SCSI on ST Virtual Connection endpoint intends to function as a SCSI Initiator, and the remote endpoint indicates that an authentication handshake is required, the SCSI Initiator endpoint shall begin the authentication handshake as described in TBD (TODO).

If authentication is required, a SCSI on ST Virtual Connection endpoint shall not initiate any SCSI operations until the authentication handshake has been successfully performed. A SCSI on ST Target shall respond to any data transfer request (Request\_To\_Send, Request\_To\_Receive, or Request\_Zero\_Length) with a Request\_Answer with a reason code of Authentication Required (TBD) until the SCSI on ST authentication handshake is performed. A SCSI on ST Target may adopt any suitable policy for reclaiming Virtual Connection resources if authentication is required and a successful SCSI on ST authentication sequence is not performed by a SCSI on ST Initiator. Specifically, such a policy should protect against denial of service attacks by unauthorized SCSI on ST connection Initiators.

### 1.3.2 Connection Teardown

=====

SCSI on ST connections shall be torn down using the standard ST Virtual Connection teardown sequence.

### 1.3.3 Virtual Circuit Keep Alive

=====

A SCSI on ST endpoint which behaves as a SCSI Target shall implement ST Virtual Connection keep-alive sequence as described in the ST specification. This keep-alive sequence is the primary mechanism for SCSI on ST Targets to reclaim resources associated with a broken Virtual Connection.

A SCSI on ST endpoint which behaves as a SCSI Initiator may implement the ST Virtual Connection keep-alive sequence as described in the ST specification.

#### 1.4 Device management

=====

An application client begins a SCSI on ST I/O Transaction when it provides to the SCSI on ST layer a request for an Execute command service. A single request or a list of linked requests may be presented to the software interface. The SCSI on ST layer then performs the following actions using ST services to perform the SCSI command.

The ST port that is the SCSI Initiator for the command starts a Transaction by sending an ST Request\_To\_Send, ST Request\_To\_Receive, or SCSI on ST Request\_Zero\_Length (see section 2.2, ST NOP Operations in SCSI On ST) operation to begin a SCSI on ST device management sequence. A Request\_To\_Send is sent if the SCSI command involves data transfer from the SCSI Initiator to the SCSI Target. A Request\_To\_Receive is sent if the SCSI command involves data transfer from the SCSI Target to the SCSI Initiator. A SCSI on ST Request\_Zero\_Length is sent if the command involves no data transfer.

The ST Request\_To\_Send, Request\_To\_Receive or SCSI on ST Request\_Zero\_Length operation has an optional payload including some command control flags, addressing information, and the SCSI command descriptor block (CDB).

This ST Request\_To\_Send, Request\_To\_Receive or SCSI on ST Request\_Zero\_Length operation is the Execute Command service request and starts the I/O Operation. The Transaction that is started is identified by the ST Initiator Sequence Identifier, qualified by the ST Virtual Connection. The ST Initiator Sequence Identifier used on the ST Request\_To\_Send, Request\_To\_Receive, or SCSI on ST Request\_Zero\_Length operation which begins a SCSI on ST Transaction is called the Initiator Transaction Identifier and is used appropriately on all ST operations associated with the Transaction.

When the Target for the command has completed the interpretation of the command, has determined that data transfer is required, and is prepared to request the data delivery service, it indicates this to the Initiator, and data is transferred, as described below. Note that the amount of data transferred is the minimum of:

- 1) the transfer length specified in the ST Request\_To\_Send or Request\_To\_Receive operation which initiated the Transaction
- 2) the amount of data the Target requires to be transferred as specified in the SCSI CDB.

If the Transaction was initiated with an ST Request\_To\_Send operation, the Target sends one or more ST Clear\_To\_Send operations for the amount of data which it is prepared to receive.

The Initiator then responds with sequences of one or more ST Data operations to satisfy the outstanding Clear\_To\_Send operations received from the Target. The last ST Data operation in each sequence

shall have the ST Interrupt and Last flag bits set, and the ST Silent flag bit cleared.

When the Target is prepared to receive additional data, it may send one or more additional ST Clear\_To\_Send operations, to which the Initiator will respond with additional data sequences, until the entire, required amount of data has been transferred.

If the Transaction was initiated with an ST Request\_To\_Receive operation, the Target responds to the ST Request\_To\_Receive operation with an ST Request\_To\_Send operation. Data is then transferred from the Target to the Initiator as above, except with the Initiator sending ST Clear\_To\_Send operations, and the Target sending ST Data operations.

After all the data has been transferred, the Target transmits the Execute Command service response by sending an ST Put sequence to the Status Persistent Memory Region defined by the connection setup protocol, at an address (Bufx/Offset) computed from the Initiator Transaction Identifier for the Transaction. This ST Put sequence contains the SCSI status and, if an unusual condition has been detected, the SCSI REQUEST SENSE information and the SCSI on ST response information describing the condition. The SCSI on ST Status Put terminates the command. The SCSI logical unit determines whether additional commands will be performed in the SCSI on ST I/O Operation. If this is the last or only command executed in the SCSI on ST I/O Operation, the SCSI on ST I/O Operation and the Transaction are terminated.

When the command is completed, returned information is used to prepare and return the Execute Command service confirmation information to the software that requested the operation. The returned status indicates whether or not the command was successful. The successful completion of the command indicates that the SCSI device performed the desired operations with the transferred data and that the information was successfully transferred to or from the SCSI Initiator.

If the command is linked to another command, the ST Put payload shall contain the proper status indicating that another command will be executed. The Initiator shall continue the same Transaction with a SCSI on ST Request\_To\_Send or Request\_To\_Receive operation, beginning the next SCSI command. All SCSI commands linked in the SCSI on ST I/O Operation except the last are executed in the manner described above.

Note that when a SCSI on ST Transaction executes more than one linked SCSI command, the same Initiator Transaction Identifier is used for all ST Sequences in the SCSI on ST Transaction.

The number of SCSI on ST I/O Operations that may be active at one time depends on the queuing capabilities of the particular SCSI devices and the number of concurrent exchanges supported by the ST ports.

A brief way of summarizing this is that a SCSI on ST Transaction

consists of one pair of:

- 1) an ST Scheduled Transfer (read or write) sequence
- 2) an ST Put sequence with status

for each (linked) command in a single SCSI Execute service request.

#### 1.4.1 Underrun Residual Handling

=====

If the amount of data the Target requires to be transferred as specified in the SCSI CDB is less than the length of the ST data transfer specified in the ST Request\_To\_Send or Request\_To\_Receive operation, either as a result of a successful or unsuccessful SCSI command completion, this is called an underrun residual condition.

If an ST Write sequence in a SCSI on ST Transaction results in an underrun residual, the SCSI on ST Target shall receive all data it has requested with outstanding ST Clear\_To\_Send operations before performing the SCSI on ST Status Put operation which signifies the completion of the SCSI command.

If an ST Read sequence in a SCSI on ST Transaction results in an underrun residual, and the SCSI on ST Initiator has sent ST Clear\_To\_Send operations for this ST Read sequence for data beyond the final block in which the SCSI on ST Target has transferred data, the SCSI on ST Initiator shall ensure that all Clear\_To\_Send operations have been received by the SCSI on ST Target by sending an ST End operation, with an optional payload that indicates the highest block number of Clear\_To\_Send operations which have been sent (see 2.5). The SCSI on ST Target shall respond to the ST End operation with an ST End\_Ack operation as described by the ST protocol, after all outstanding Clear\_To\_Send operations have been received. In order to support this Clear\_To\_Send accounting protocol, a SCSI on ST Initiator shall only send Clear\_To\_Send operations for a contiguous sequence of Blocks for SCSI on ST Read Transactions.

#### 1.4.2 Third Party SCSI Commands

=====

Certain third-party SCSI commands and parameters specify a 64-bit field that is defined to access other SCSI devices addressable from that port. These commands include COPY, RESERVE, and several others.

The ST protocol does not specify any form of addressing. However, it is usually run on a Lower Level Protocol which does specify an address format. Therefore, addressing format for third-party SCSI commands in the SCSI on ST protocol is a function of the Lower Level Protocol addressing format.

These addressing formats are beyond the scope of this specification.

#### 1.5 Task Management

=====

An application client requests a task management function when a task or some group of tasks must be aborted or terminated.

SCSI task management functions are mapped on to SCSI on ST and its underlying ST as:

SCSI	SCSI on ST	ST	Required
ABORT TASK	End sequence	End sequence	Y
TERMINATE TASK	Request_Zero_Length with Terminate Task bit set	NOP operation	N
TARGET RESET	Transaction with Target Reset bit set	NOP operation and Put operation	Y
ABORT TASK SET	Transaction with Abort Task Set bit set	NOP operation and Put operation	Y
CLEAR TASK SET	Transaction with Clear Task Set bit set	NOP operation and Put operation	Y
CLEAR ACA	Transaction with Clear Auto Contingent Allegiance bit set	NOP operation and Put operation supported	Y if ACA is

The SCSI on ST Request\_Zero\_Length operation sent to initiate a TARGET RESET, ABORT TASK SET, CLEAR TASK SET or CLEAR ACA task management function is sent first operation in a new Transaction. As with all other Transactions, the target shall respond TARGET RESET, ABORT TASK SET, CLEAR TASK SET and CLEAR ACA task management functions with a SCSI on ST Status Put using the Initiator Transaction Identifier specified in the Request\_Zero\_Length operation that requested the task mangement function.

The SCSI on ST Request\_Zero\_Length operation sent to initiate a TERMINATE TASK task management function is sent using the Initiator Transaction Identifier of the task to be terminated. Unlike other task management functions initiated with a SCSI on ST Request\_Zero\_Length operation, the target will not respond to a TERMINATE TASK task management request with a SCSI on ST Status Put operation specifically for the TERMINATE TASK task management function. However, when the requested task is terminated, the target shall, as always, perform a SCSI on ST Status Put operation with the status of the terminated task.

## 2 SCSI On ST Protocol Operation Formats

=====

### 2.1 Optional Payload for Connection\_Request and Connection\_Answer

=====

Field Name	Description	Size
TYPE	ST optional payload type code	1 byte
LENGTH	ST optional payload size field	1 byte
MAX_TARGET	Maximum supported target address	2 bytes
FLAGS	SCSI on ST feature flags	4 bytes
SPMR_SIZE	Status Persistent Memory Region Size	4 bytes

2.1.1 TYPE  
=====

Always 3, indicating that the ST optional payload field is valid and contains a ULP parameter

2.1.2 LENGTH  
=====

Always 32, indicating that the ST optional payload field is 32 bytes long.

2.1.3 MAX\_TARGET  
=====

Maximum supported value of the TARGET field (see section 2.3.3) in the optional payload for ST Request\_To\_Send and Request\_To\_Receive operations.

2.1.4 FLAGS  
=====

The FLAGS field contains a number of supported and required feature flags:

Bit Definition	Bit
-----	
Byte 0 (most significant)	
RECEIVE_ALIGN	7-5
SEND_ALIGN	4-0
Byte 1	
reserved	7-0
Byte 2	
reserved	7-2
ONE_CTS_READ	0
Byte 3	
reserved	7-5
DATA_CHANNEL	4-3
AUTHENTICATE	2
TARGET	1
INITIATOR	0

INITIATOR indicates that the SCSI on ST endpoint is capable of performing SCSI Initiator functions.

TARGET indicates that the SCSI on ST endpoint is capable of performing SCSI Target functions.

AUTHENTICATE indicates that an authentication handshake must be performed during a SCSI on ST connection setup before data may be transferred.

DATA\_CHANNEL indicates the ST Data Channel for which to request the SCSI on ST Status Persistent Memory Region, and on which all SCSI on ST Data operations, including SCSI on ST Status Put sequence shall be performed.

ONE\_CTS\_READ indicates that a SCSI on ST Target requires an Initiator to send only single ST Clear\_To\_Send operation to cover the entire data transfer of a SCSI read operation.

SEND\_ALIGN is the log, base 2, of the minimum required alignment of the offset parameter of any ST Clear\_To\_Send operations sent to the SCSI on ST Virtual Connection endpoint specifying the SEND\_ALIGN value.

RECEIVE\_ALIGN is the log, base 2, of the minimum required alignment of the offset parameter of any ST Data operations sent to the SCSI on ST Virtual Connection endpoint specifying the RECEIVE\_ALIGN value. This, necessarily implies that a SCSI on ST endpoint will never perform a Clear\_To\_Send operation with an initial offset which does not conform to the RECEIVE\_ALIGN value it specified during the SCSI on ST connection setup.

#### 2.1.5 SPMR\_SIZE =====

This field indicates the number of 512 byte Status Persistent Memory Region segments a SCSI on ST Target should request that a SCSI on ST Initiator should expose.

In other words, the size of the Status Persistent Memory Region requested shall be:

$$\text{SPMR\_SIZE} * 512 \text{ bytes}$$

#### 2.1.6 SCSI on ST Virtual Connection Reject Reason Codes =====

If the parameters of a SCSI on ST Virtual Circuit are unacceptable, the Virtual Circuit request shall be rejected, or closed with an appropriate reason code:

Definition	Value (hex)
-----	-----
Single CTS Read unsupported	TBD
Target function unsupported	TBD
Busy (no resources)	TBD

## 2.2 ST NOP Operations In SCSI On ST

=====

SCSI on ST distinguishes different uses of the ST NOP operation using the 16 bit opaque Param field as an operation code:

Definition	Value (hex)
-----	-----
Request_Zero_Length	0

The 32 bit opaque S\_id field of a SCSI on ST Request\_Zero\_Length operation contains the Initiator Transaction Identifier.

The remainder of the opaque fields SCSI on ST Request\_Zero\_Length operation are unused. The optional payload of the Request\_Zero\_Length operation contains the optional payload for SCSI on ST command operations as described in 2.3.

## 2.3 Optional Payload For SCSI on ST Command Operations

=====

The optional payload for Request\_To\_Send, Request\_To\_Receive, and Request\_Zero\_Length operations is organized as:

Field Name	Description	Size
-----	-----	-----
TYPE	ST optional payload type code	1 byte
LENGTH	ST optional payload size field	1 byte
TARGET	Attached Target address	2 bytes
LUN	Logical Unit Number	8 bytes
CNTL	Control Field	4 bytes
CDB	SCSI command descriptor block	16 bytes

### 2.3.1 TYPE

=====

Always 3, indicating that the ST optional payload field is valid and contains a ULP parameter

### 2.3.2 LENGTH

=====

Always 32, indicating that the ST optional payload field is 32 bytes long.

### 2.3.3 TARGET

=====

Selects one of a set of individual SCSI Targets addressable through a single SCSI on ST Virtual Connection. The maximum allowable value of the TARGET field is established in the SCSI on ST connection setup protocol.

#### 2.3.4 LUN

=====

The SCSI on ST Logical Unit Number (LUN) is the address of the desired logical unit in the attached subsystem. The LUN field is specified by X3.230-199X for all IUs of Category 6.

Each SCSI on ST Target shall accept an INQUIRY command to the logical unit addressed by the LUN zero (0000 0000 0000 0000 hexadecimal). Using the SCSI INQUIRY information, the Initiator can determine the SCSI device type, manufacturer, and model of the logical unit. If the logical unit at LUN 0 specifies a device model that has a defined addressing structure, the Initiator can use that information to determine what other logical units are implemented in the Target. The structure of the LUN field is not specified by the SCSI on ST protocol. The structure is specified by the SCSI device model and may be vendor unique for device models that do not have a defined address structure.

If the LUN address locates a valid logical unit, the command shall be executed according to standard SCSI behavior. Behavior may include successful execution of the command, presentation of errors associated with the command, or rejection of the command. If the addressed logical unit does not exist, the responses shall follow the SCSI-3 rules for selection of invalid logical units.

#### 2.3.4 CNTL

=====

The CNTL field contains a number of control flags and control bits:

Bit Definition	Bit
-----	
Byte 0 (most significant), reserved	
reserved	7-0
Byte 1, Task codes	
reserved	7-3
TASK ATTRIBUTE	2-0
Byte 2, Task management flags	
TERMINATE TASK	7
CLEAR ACA	6
TARGET RESET	5
reserved	4-3
CLEAR TASK SET	2
ABORT TASK SET	1
reserved	0
Byte 3, Execution management codes	
reserved	7-2
READ DATA	1
WRITE DATA	0

##### 2.3.4.1 Task Codes, Byte 1

=====

TASK ATTRIBUTE can be:

Value, bits 2-0	TASK ATTRIBUTE
000 b	SIMPLE_Q
001 b	HEAD_OF_Q
010 b	ORDERED_Q
100 b	ACA_Q
101 b	UNTAGGED
others	reserved

SIMPLE\_Q requests that the task be managed according to the rules for a SIMPLE task attribute.

HEAD\_OF\_Q requests that the task be managed according to the rules for a HEAD OF QUEUE task attribute.

ORDERED\_Q requests that the task be managed according to the rules for an ORDERED task attribute.

ACA\_Q requests that the task be managed according to the rules for an automatic contingent allegiance (ACA) task attribute.

UNTAGGED requests that the task be managed according to the rules for an untagged task. Only one untagged task can exist for each logical unit identifier / Initiator pair. Requesting a second untagged command for the same logical unit identifier / Initiator pair shall be treated as an overlapped command. [X3.270-199X] An untagged task is scheduled according to the rules for a SIMPLE task attribute.

#### 2.3.4.2 Task Management Flags, Byte 2

=====

If any task management flag is set to 1, the CDB, and CDB related CNTL flags (task codes and execution management codes) are not valid and are ignored. No more than one task management flag shall be set to 1 in any ST Request\_To\_Send or Request\_To\_Receive operation.

TERMINATE TASK, when set to 1, requests that the specified task be terminated without corrupting the medium. This bit is optional for SCSI on ST Targets. The TERMINATE TASK operation shall be sent by the initiator using the Initiator Transaction Identifier for the task to be terminated.

CLEAR ACA, when set to 1, causes the ACA condition to be cleared. When the task manager clears the auto contingent allegiance condition, any task within that task set may be completed subject to the rules for task management specified by X3.270-199X.

The use of the ACA bit in the CDB control field and the implementation of ACA is described in X3.270-199X.

If the ACA bit in the CDB control field is set to 0, the automatic sense operation performed by the presentation of the SCSI on ST Status

Put operation shall clear the ACA condition. Depending on the MODE SELECT parameters that have been established, additional SCSI on ST I/O operations may have to be aborted by the recovery abort

TARGET RESET, when set to 1, performs a reset to the SCSI device as defined in X3.270-199X. TARGET RESET resets all tasks for all Initiators and resets all internal states of the Target to their initial power on and default values. A unit attention condition is created for all Initiators. The Initiator and Target clear all resources.

CLEAR TASK SET causes all tasks from all Initiators in the specified task set to be aborted as defined in X3.270-199X. A unit attention condition is created for all Initiators other than the Initiator that sent the CLEAR TASK SET that had tasks in the task set.

ABORT TASK SET causes all tasks in the task set from the Initiator requesting the ABORT TASK SET to be aborted as defined in X3.270-199X. The bit is mandatory in SCSI on ST.

#### 2.3.4.3 Execution management codes, Byte 3 =====

READ DATA may only be set on an ST Request\_To\_Receive operation. When set to 1, specifies that the Initiator expects data for the task to be in the direction opposite to the direction of the ST Request\_To\_Receive operation. This is a SCSI read-type operation.

WRITE DATA may only be set on an ST Request\_To\_Send operation. When set to 1, specifies that the Initiator expects data for the task to be in the same direction as the ST Request\_To\_Send operation. This is a SCSI write operation.

If both READ DATA and WRITE DATA are set to 0, there shall be no data and ST Request\_To\_Send or Request\_To\_Receive transfer length shall be 0.

#### 2.3.5 CDB =====

The CDB field contains the actual CDB to be interpreted by the addressed logical unit. The maximum CDB length is 16 bytes. The CDB is not valid and is ignored if any task management flag is set to 1. The command byte of the CDB shall be located at the first byte transmitted of the CDB field. The CDB's first byte defines the length and basic format of the remainder of the CDB as described in X3.270-199X. Bytes beyond the last byte of the CDB are not defined by SCSI on ST, shall be ignored by the Target, and may have any value.

#### 2.4 SCSI on ST Status Put Sequence =====

The SCSI on ST Status Put sequence shall be directed to an address

(bufx/offset) which is computed as:

```

address = (Status Persistent Memory Region initial bufx * Initiator bufsize
          + Status Persistent Memory Region initial offset)
          + 512 * Transaction Initiator Identifier
Status Put bufx = address / Initiator bufsize
Status Put offset = address % Initiator bufsize

```

In addition, the Transaction Initiator Identifier is sent in the ST S\_id field, defined as opaque in the ST specification, of the ST Data operations(s) of the SCSI on ST Status Put. This is a departure from all other ST operations for a Transaction, where the Target puts the Initiator Transaction Identifier in the ST D\_id field. The D\_id field of SCSI on ST Status Put operations contain the Responder Identifier for the Persistent Memory Region identified in the ST Memory\_Region\_Available operation received as part of the SCSI on ST connection setup protocol.

The SCSI on ST Status Put sequence shall consist of a single ST Data STU, with the entire SCSI on ST Status Put payload. The SCSI on ST Status Put Data STU shall have the ST Interrupt and Last flags bits set, and the ST Silent flag bit cleared.

The payload of the SCSI on ST Status Put sequence is:

Field Name	Description	Size
reserved		8 bytes
STATUS	Field validity and SCSI status	4 bytes
RESID	Residual count	4 bytes
SNS_LEN	Length of SNS_INFO field	4 bytes
RSP_LEN	Length of RSP_INFO field	4 bytes
RSP_INFO	SCSI on ST response information	m bytes (0, 4 or 8)
SNS_INFO	SCSI sense information	n bytes

#### 2.4.1 STATUS =====

The STATUS field is normally 0 upon successful completion of a SCSI on ST I/O Operation. If command linking is being performed, an SCSI on ST Status Put is performed for each command executed. A value of 0 means no error, and in addition indicates that no other information is present in the SCSI on ST Status Put. For linked commands, INTERMEDIATE status or INTERMEDIATE CONDITION MET status indicates successful completion of a command with no other information valid if all other fields are 0.

The format of the STATUS field is:

Bit Definition	Bit
Byte 0 (most significant)	
reserved	7-0
Byte 1	

reserved	7-0
Byte 2	
reserved	7-4
RESID_UNDER	3
RESID_OVER	2
SNS_LEN_VALID	1
RSP_LEN_VALID	0
Byte 3	
SCSI status byte	7-0

RESID\_UNDER indicates that the RESID field is valid and contains the count of bytes that were expected to be transferred, but were not transferred.

RESID\_OVER indicates that the RESID field is valid and contains the count of bytes that could not be transferred because the ST transfer length was not sufficient.

SNS\_LEN\_VALID indicates that the SNS\_LEN field is valid and contains the count of bytes in the SNS\_INFO field.

RSP\_LEN\_VALID indicates that the RSP\_LEN field is valid and contains the count of bytes in the RSP\_INFO field.

Byte 3 (LSB) contains the status byte from the SCSI logical unit. The status byte codes are defined by X3.270-199X.

#### 2.4.2 RESID =====

If the RESID\_UNDER or the RESID\_OVER bit is 1, the RESID field contains a count of the number of residual data bytes which were not transferred in the DATA IUs for this SCSI command. Upon successful completion of a SCSI on ST I/O Operation, the residual value is normally 0 and the RESID value is not valid. Devices having indeterminate data lengths may have a nonzero residual byte count after completing valid operations. Targets are not required to verify that the data length implied by the contents of the CDB will create an overrun or underrun before beginning execution of an SCSI command.

If the RESID\_UNDER bit is set, a transfer that did not fill the buffer to the expected displacement, specified by the ST transfer length, was performed and the value of RESID is a number equal to:

$$\text{ST transfer length} - \text{highest offset of any byte transmitted}$$

A condition of RESID\_UNDER may not be an error for some devices and some commands.

If the RESID\_OVER bit is set, the transfer was truncated because the data transfer required by the SCSI command extended beyond the displacement value of DL. Those bytes that could be transferred without violating the ST transfer length value may be transferred. The RESID is a number equal to:

(Transfer length required by command) - ST transfer length

If a condition of RESID\_OVER is detected, the termination state of the SCSI on ST I/O operation is not certain. Data may or may not have been transferred and the SCSI status byte may or may not provide correct command completion information.

If the RESID\_UNDER and the RESID\_OVER bits are 0, the RESID field is not meaningful and may have any value.

#### 2.4.3 SNS\_LEN =====

If the SNS\_LEN\_VALID bit is 1, the SNS\_LEN field specifies the number of valid bytes of SNS\_INFO.

If the SNS\_LEN\_VALID bit is 0, the SNS\_LEN field is not valid and no SNS\_INFO is provided.

The SNS\_LEN field is always included in the SCSI on ST Status Put.

#### 2.4.4 RSP\_LEN =====

If the RSP\_LEN\_VALID bit is 1, the RSP\_LEN field specifies the number of valid bytes of RSP\_INFO. The number shall be 0, 4, or 8. Other values of length are reserved for future standardization.

An RSP\_LEN of 0 specifies that no bytes of response information are being provided.

If the RSP\_LEN\_VALID bit is 0, the RSP\_LEN field is not valid and no RSP\_INFO is provided.

The RSP\_LEN field is always included in the SCSI on ST Status Put.

#### 2.4.5 RSP\_INFO =====

The RSP\_INFO field contains information describing only the protocol failures detected during the execution of a SCSI on ST I/O operation. The RSP\_INFO does not contain SCSI logical unit error information, since that is contained in the SNS\_INFO field as described in 2.4.6. The RSP\_INFO field shall contain valid information if the Target detects any of the conditions indicated by a RSP\_CODE. The format of the RSP\_INFO field is:

Bit Definition	Bit
-----	
Byte 0 (most significant)	
reserved	7-0
Byte 1	
reserved	7-0

Byte 2		
reserved		7-0
Byte 3		
RSP_CODE		7-0

The valid RSP\_CODE values are:

RSP_CODE definition	Value (hexadecimal)
-----	-----
No failure or Task Management function complete	0
reserved	1
SCSI on ST CTS or RTS payload fields invalid	2
reserved	3
Task Management Function Not Supported	4
Task Management Function Failed	5
Nonexistent Target	6
Busy (no resources)	7

#### 2.4.6 SNS\_INFO

=====

The SNS\_INFO field contains the information specified by X3.131-1994 for presentation by the REQUEST SENSE command. The proper SNS\_INFO shall be presented when the SCSI status byte of CHECK CONDITION or COMMAND TERMINATED is presented as specified by X3.270-199X. SCSI on ST devices should always use autosense.

#### 2.5 Optional Payload For ST End And End\_Ack Operations

=====

When the ST End or End\_Ack operation is required by the SCSI on ST protocol to specify the block number of the last ST Clear\_To\_Send operation sent, the ST End or End\_Ack operation's optional payload shall be:

Field Name	Description	Size
-----	-----	-----
TYPE	ST optional payload type code	1 byte
LENGTH	ST optional payload size field	1 byte
reserved		2 bytes
B_NUM	Block number of last CTS sent	4 bytes

##### 2.5.1 TYPE

=====

Always 3, indicating that the ST optional payload field is valid and contains a ULP parameter.

##### 2.5.2 LENGTH

=====

Always 8, indicating that the ST optional payload field is 8 bytes

long.

### 2.5.2 B\_NUM

=====

Indicates the ST Block number of the last ST Clear\_To\_Send operation sent by the data sink.

Since the SCSI on ST protocol requires in order request of ST data Blocks, the ST Block number of the last ST Clear\_To\_Send operation permits the data source to ensure that all ST Clear\_To\_Send operations for a Transaction have been received before reusing the ST Sequence Identifiers associated with the Transaction. This permits the endpoints to avoid aliasing of ST Clear\_To\_Send operations across Transactions when Sequence Identifiers are reused.

## 3 ERROR RECOVERY PROCEDURES

=====

### 3.1 Virtual Circuit Keep Alive

=====

A SCSI on ST endpoint may implement ST Virtual Connection keep-alive sequences as described in the ST specification.

If an ST Virtual Connection is discovered to be dead by a failure of the keep-alive test, all SCSI on ST Transactions on the Virtual Circuit shall be declared dead and their resources reclaimed, as well as the resources for the Virtual Connection itself.

### 3.2 Operation Timeouts

=====

A SCSI on ST Initiator shall maintain timeouts on all outstanding device and task management operations from the time a Request\_To\_Send, Request\_To\_Receive or Request\_Zero\_Length operation is sent until the final Data and SCSI on ST Status Put operations are received.

The operation timeout shall be at least the sum of:

- 1) the maximum time required to perform the SCSI command and transfer the associated data
- 2) the maximum time an ST operation can remain in the network

This timeout ensures that no ST operations remain in the network or will be subsequently generated by a SCSI on ST Target for a SCSI on ST Transaction, which guarantees that a SCSI on ST Initiator may reuse the I-id associated with the SCSI on ST Transaction immediately.

If an operation timeout expires, the Initiator shall first determine the viability of the SCSI on ST Virtual Connection, with an ST Request\_State/Request\_State\_Response sequence. If a Request\_State\_Response operation is not received within an appropriate period of time, the Request\_State/Request\_State\_Response sequence

shall be retried an appropriate number of times, after which the Virtual Connection shall be declared dead, and its resources reclaimed.

If a Request\_State\_Response is received, the operation which timed out shall be aborted with an ST End sequence. If an ST End\_Ack operation is not received within the maximum an ST operation can remain in the network, the ST End sequence shall be retried an appropriate number of times, after which the SCSI on ST Transaction shall be declared dead and its resources reclaimed.

A SCSI on ST Target shall not maintain timeouts on outstanding Transactions. A SCSI on ST Target's resources for a Transaction shall only be reclaimed when:

- 1) the Target completes the Transaction
- 2) the Initiator ends the Transaction with an ST End operation
- 3) the SCSI on ST Virtual Connection is ended

The requirement that the SCSI on ST Initiator perform an ST End sequence for timed out operations ensures that a SCSI on ST Target can deterministically reclaim resources associated with a failed Transaction.

UPDATED

=====

Replaced all uses of device server with Target.

Capitalized Transaction, Initiator and Target, Status Put

Replaced the name ``SCSI on ST Zero\_Length\_Command`` with ``SCSI on ST Request\_Zero\_Length``

Added TERMINATE TASK SCSI task management function.

Completed End and End\_Ack optional payload definition.

Added the requirement that out of order Block support not be requested during connection setup. In order Blocks are required for the CTS accounting protocol.

Added definition of opaque S\_id field as Initiator Transaction Identifier in SCSI on ST Request\_Zero\_Length operation.

\$Id: scsiOnSt.txt,v 1.11 1999/04/09 22:39:55 steph Exp steph \$