

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 operations (e.g. read operation, write operation, PMR put operation), which are logically associated to implement a complete function from the list above.

The correspondence between SCSI functions and elements of the SCSI on ST protocol are:

SCSI	SCSI on ST
I/O Operation	Transaction
Command Service Request	Request_To_Send or Request_To_Receive message
Data delivery request	Clear_To_Send message
Data delivery action	Data message (STU)
Command service response	Put sequence

## 1.2 Common ST Message Characteristics

=====

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

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

All ST Data messages shall be sent on ST Data Channel 3. ST Data Channel flag bits shall be appropriately set to reflect this.

## 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 established by sending an ST Request\_Connect message to the well known port for SCSI on ST (TBD), with SCSI on ST connection parameters in the optional payload.

If the connection request is acceptable to the SCSI target, it responds with an ST Connection\_Answer message, with its SCSI on ST connection parameters in the optional payload.

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, with an ST Request\_Memory\_Region message. The connection initiator shall expose the Status Persistent Memory Region and respond with an appropriate Memory\_Region\_Available message. The connection responder will record the parameters of the Status Persistent Memory Region with the Virtual Connection state.

When the Connection\_Answer message 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 end point shall not initiate any SCSI operations until the Status Persistent Memory Region has been

requested, exposed and the ST Memory\_Region\_Available message sent. A SCSI on ST initiator shall keep a timer awaiting the ST Request\_Memory\_Region message, 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 data transfer request (Request\_To\_Send or Request\_To\_Receive) with a Request\_Answer with a reason code of Status Persistent Memory Region Not Established (TBD) until the ST Memory\_Region\_Available message is received.

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.

(TODO: this doesn't support any way for the initiator to negotiate down the responder).

### 1.3.2 Connection Teardown

=====

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

### 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. Each request contains all the information necessary for the execution of one SCSI command, including the local storage address and characteristics of data to be transferred by the command. 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, or an ST Request\_To\_Receive message to begin an ST data transfer sequence. A Request\_To\_Send is sent if the SCSI command involves data transfer from the SCSI initiator to the SCSI target, or no data transfer. A Request\_To\_Receive is sent if the SCSI command involves data transfer from the SCSI target to the SCSI initiator.

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

This ST Request\_To\_Send or Request\_To\_Receive message is the Execute

Command service request and starts the I/O Operation. The transaction that is started is identified by the ST Initiator ID, qualified by the ST port (Virtual Connection) identifier. This Initiator ID is set appropriately on all ST messages associated with the transaction.

When the device server 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 message which initiated the transaction
- 2) the amount of data the device server requires to be transferred as specified in the SCSI CDB.

If the transaction was initiated with an ST Request\_To\_Send message, the device server sends one or more ST Clear\_To\_Sends for the amount of data which it is prepared to receive.

The initiator then responds with sequences of one or ST Data messages satisfy the outstanding Clear\_To\_Sends received from the device server. The last ST Data message in each sequence will have the ST Interrupt and Last flag bits set, and the ST Silent flag bit cleared.

When the device server is prepared to receive additional data, it may send one or more additional ST Clear\_To\_Sends, 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 message, the device server ``echos'' the ST Request\_To\_Receive message back to the initiator as an ST Request\_To\_Send message. Data is then transferred from the device server to the initiator as above, except with the initiator sending ST Clear\_To\_Send messages, and the device server sending ST Data message sequences.

After all the data has been transferred, the device server transmits the Execute Command service response by sending an ST Put message to the Status Persistent Memory Region defined by the connection setup protocol, at an address (bufx/offset) computed from the Initiator ID for the transaction. This ST Put message contains the SCSI status and, if an unusual condition has been detected, the SCSI REQUEST SENSE information and the FCP response information describing the condition. The command 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. A protocol is provided to present error information if a command is not successful.

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 message, 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.

The number of FCP 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.

If successful or failed execution of a SCSI command started with an ST Request\_To\_Send message results in no data transfer, the device server shall perform an immediate SCSI on ST status Put operation, without sending any Clear\_To\_Send messages. If an error occurs on a SCSI command started with an ST Request\_To\_Send message after one or more Clear\_To\_Sends have been sent, the data transfers for these Clear\_To\_Sends shall be completed before the device server performs a SCSI on ST status Put operation.

If successful or failed execution of a SCSI command started with an ST Request\_To\_Receive message results in no data transfer, the device server may either perform an immediate SCSI on ST status Put operation, or may send the Request\_To\_Send message and then perform an ST status Put operation. If an error occurs on a SCSI command started with an ST Request\_To\_Receive message after some data has been transferred, the device server may perform a SCSI on ST status put operation at any time. The device server shall ignore ST Data and Clear\_To\_Send messages for inactive transactions, or transactions for which no SCSI command is currently being performed.

A brief way of summarizing this is that a SCSI on ST transaction consists of a pairs of:

- 1) an ST data transfer operation
- 2) an ST Put operation with status

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

#### 1.4.1 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 must 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.

Some SCSI task management functions are mapped into ST control sequences and others are mapped into control bits in the SCSI on ST optional payload of an ST Request\_To\_Send message, as part of a complete transaction.

Task management functions that use the ST Request\_To\_Send message are sent as the first message in a new transaction. Task management functions which are mapped to a SCSI on ST transaction end with a SCSI on ST status Put that indicates whether it was correctly accepted. The CDB field in a SCSI on ST optional payload that perform task management functions is ignored. The mappings are:

SCSI	SCSI on ST
TARGET RESET	SCSI on ST transaction with target reset
bit	set
ABORT TASK	ST End sequence
ABORT TASK SET	SCSI on ST transaction with Abort Task Set bit set
CLEAR TASK SET	SCSI on ST transaction with Clear Task Set bit set
CLEAR ACA	SCSI on ST transaction with Clear Auto Contingent Allegiance bit set (only ACA is supported by target)

Note that this mapping expands the impact of the ST End sequence from terminating a single data transfer to terminating an entire SCSI on ST transaction.

## 2 SCSI On ST Protocol Message 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 2.2.3) in the optional payload for ST Request\_To\_Send and Request\_To\_Receive messages.

### 2.1.4 FLAGS

=====

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

Bit Definition	Bit
-----	
Byte 0 (most significant)	
reserved	7-0
Byte 1	
reserved	7-0
Byte 2	
reserved	7-1
SPMR_ALIGN	1
ONE_CTS_READ	0
Byte 3	
reserved	7-2
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.

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

SPMR\_ALIGN indicates that a SCSI on ST target requires that an initiator expose a Status Persistent Memory Region which is aligned on multiples of the Status Persistent Memory Region segment size, 512 bytes.

### 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:

SPMR\_SIZE \* 512 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
Aligned SPMR unsupported	TBD
Target function unsupported	TBD
Busy (no resources)	TBD

### 2.2 Optional Payload For Request\_To\_Send and Request\_To\_Receive

=====

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

#### 2.2.1 TYPE

=====

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

#### 2.2.2 LENGTH

=====

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

#### 2.2.3 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.

#### 2.2.4 LUN

=====

The FCP logical unit number is the address of the desired logical unit in the attached subsystem. The FCP\_LUN field is specified by X3.230-199X for all IUs of Category 6.

Each SCSI FCP target shall accept an INQUIRY command to the logical unit addressed by the FCP\_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 FCP\_LUN field is not specified by the FCP. 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. An example of a four-layer hierarchical address is given in annex C.

If the FCP\_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.

2.2.4 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	
reserved	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

2.2.4.1 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.

#### 2.2.4.2 Task Management Flags, Byte 2

=====

If any task management flag is set to 1, the CDB, and CDB related FCP\_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 message.

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 FCP\_RSP IU 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.

#### 2.2.4.3 Execution management codes, Byte 3

=====

READ DATA may only be set on an ST Request\_To\_Receive message. 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 message. This is a SCSI read-type operation.

WRITE DATA may only be set on an ST Request\_To\_Send message. 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 message. 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.

#### 2.2.5 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.

#### 2.3 SCSI on ST Status Put Operation

=====

The SCSI on ST Status Put operation 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 ID
Status Put bufx = address / initiator bufsize
Status Put offset = address % initiator bufsize
```

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

The last message of a SCSI on ST Status Put sequence shall have the ST Interrupt and Last flags bits set, and the ST Silent flag bit cleared.

The content 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

### 2.3.1 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 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 FCP\_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.

### 2.3.2 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:

ST transfer length - 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.

### 2.3.3 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.

### 2.3.4 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.

### 2.3.5 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 2.2.6. The RSP\_INFO field shall contain valid information if the target detects any of the conditions indicated by an FCP 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

### 2.3.6 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.

## 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.

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 message 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 message is not received within an appropriate amount of time, 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 operations. A SCSI on ST target's resources for an operation shall not be reclaimed unless the SCSI on ST Virtual Connection is ended. However, a SCSI on ST target may force an operation to be terminated using an ST End sequence, if necessary.

\$Id: scsiOnSt.txt,v 1.2 1999/01/06 00:01:29 steph Exp steph \$