dpANS Project
American National Standard

T10/2229-D

Revision 0
November 9, 2010

Information technology - USB Attached SCSI - 2 (UAS-2)

T10 Technical Editor:

Curtis E. Stevens
Western Digital Technologies, Inc.
20511 Lake Forest Dr.
Lake Forest, Ca. 92630
USA

Telephone: 949-672-7933
Email: Curtis.Stevens@wdc.com

Reference number
ISO/IEC xxxx-xxx:201x
ANSI INCITS ***-201x

Points of Contact

International Committee for Information Technology Standards (INCITS) T10 Technical Committee

T10 Chair
John B. Lohmeyer
LSI Corporation
4420 ArrowsWest Drive
Colorado Springs, CO 80907-3444
USA

Telephone: (719) 533-7560
Email: lohmeyer@t10.org

T10 Vice-Chair
Mark S. Evans
Western Digital Corporation
5836 Rue Ferrari
San Jose, CA 95138
USA

Telephone: (408) 363-5257
Email: mark.evans@wdc.com

T10 Web Site: http://www.t10.org

T10 E-mail reflector:
Server: majordomo@t10.org
To subscribe send e-mail with ‘subscribe’ in message body
To unsubscribe send e-mail with ‘unsubscribe’ in message body

INCITS Secretariat
Suite 200
1250 Eye Street, NW
Washington, DC 20005
USA

Telephone: 202-737-8888
Web site: http://www.incits.org
Email: incits@itic.org

Information Technology Industry Council
Web site: http://www.itic.org

Document Distribution
INCITS Online Store
managed by Techstreet
1327 Jones Drive
Ann Arbor, MI 48105
USA

Web site: http://www.techstreet.com/incits.html
Telephone: (734) 302-7801 or (800) 699-9277

Global Engineering Documents, an IHS Company
15 Inverness Way East
Englewood, CO 80112-5704
USA

Web site: http://global.ihs.com
Telephone: (303) 397-7956 or (303) 792-2181 or (800) 854-7179

American National Standard
for Information Technology

USB Attached SCSI - 2

Secretariat
Information Technology Industry Council

Approved mm.dd.yy

American National Standards Institute, Inc.

ABSTRACT

This standard specifies the requirements for the USB Attached SCSI - 2 (UAS-2) transport protocol. The UAS-2 transport protocol defines a mechanism to transport SCSI commands using USB hardware. The UAS-2 transport protocol coordinates with other members of the SCSI family of standards via the SAM-4 architecture model. This standard is intended to be used in conjunction with SCSI command set standards and USB specifications.

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer. Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that effort be made towards their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give interpretation on any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken periodically to reaffirm, revise, or withdraw this standard. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

CAUTION: The developers of this standard have requested that holders of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are not required to avoid infringement in the use of this standard.

Published by
American National Standards Institute
11 W. 42nd Street, New York, New York 10036

Printed in the United States of America

Document Status

Revision History		
Rev	**Date**	**Description**
0	November 10, 2010	1) Initial revision. 2) Based on UAS 3) Updated the abstract, project number, and title.

New Capabilities added to UAS

Integrated Proposal List		
#	**Doc**	**Description**
1		
2		
3		
4		
5		

Contents

Tables

Figures

Foreword

This foreword is not part of American National Standard INCITS ***-201x.

The purpose of this standard is to define requirements for the transmission of SCSI commands, in a manner compliant with SAM-4, across a USB physical interface.

With any technical document there may arise questions of interpretation as new products are implemented. INCITS has established procedures to issue technical opinions concerning the standards developed by INCITS. These procedures may result in SCSI Technical Information Bulletins being published by INCITS.

These Bulletins, while reflecting the opinion of the Technical Committee that developed the standard, are intended solely as supplementary information to other users of the standard. This standard, ANSI INCITS ***-201x, as approved through the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of these Technical Information Bulletins.

Current INCITS practice is to make Technical Information Bulletins available through:

INCITS Online Store	http://www.techstreet.com/INCITS.html
managed by Techstreet	Telephone: 1-734-302-7801 or
1327 Jones Drive	1-800-699-9277
Ann Arbor, MI 48105	Facsimile: 1-734-302-7811

or

Global Engineering	http://global.ihs.com/
15 Inverness Way East	Telephone: 1-303-792-2181 or
Englewood, CO 80112-5704	1-800-854-7179
	Facsimile: 1-303-792-2192

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, National Committee for Information Technology Standards, Information Technology Institute, 1250 Eye Street, NW, Suite 200, Washington, DC 20005- 3922.

This standard was processed and approved for submittal to ANSI by the InterNational Committee for Information Technology Standards (INCITS). Committee approval of the standard does not necessarily imply that all committee members voted for approval. At the time of it approved this standard, INCITS had the following members:

The INCITS Technical Committee T10 on SCSI Storage Interfaces, that reviewed this standard, had the following members:

John B. Lohmeyer, Chair Mark Evans, Vice-Chair Ralph O. Weber, Secretary

---

Editor's Note 1: Insert list of organizations and representatives here

---

Introduction

The USB Attached SCSI standard (UAS) is divided into the following clauses:

Clause 1 describes the scope.
Clause 2 provides normative references for the entire standard.
Clause 3 provides definitions, abbreviations, and conventions used within the entire standard.
Clause 4 describes the model.
Clause 5 describes USB requirements.
Clause 6 describes transport requirements (e.g., IU's).
Clause 7 describes the SCSI Application Layer Transport Protocol Services.
Clause 8 describes device server error handling.

American National Standard
for Information Technology –

USB Attached SCSI - 2 (UAS-2)

1 Scope

The SCSI family of standards provides for different transport protocols that define the methods for exchanging information between SCSI devices. This standard defines the methods for exchanging information between SCSI devices using a USB interconnect. Other SCSI transport protocol standards define the methods for exchanging information between SCSI devices using other interconnects.

Figure 1 shows the relationship of this standard to the other standards and related projects in the SCSI family of standards.

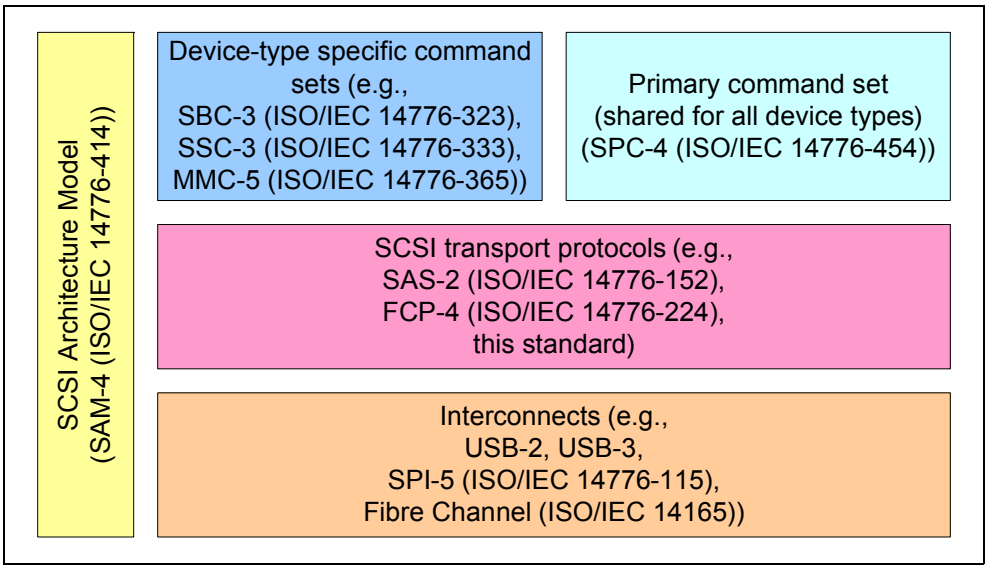


Figure 1 — SCSI document relationships

This standard describes a SCSI transport protocol (see SAM-4) for USB-2 and USB-3 with the following properties:

a) mechanism to send commands associated with any T10 command standard to a USB device;
b) complies with SCSI Architecture Model - 4 (e.g., autosense and command queuing); and
c) other capabilities.

2 Normative references

2.1 Normative references

Referenced standards and specifications contain provisions that, by reference in the text, constitute provisions of this standard. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards listed below.

Copies of the following documents may be obtained from ANSI:

a) approved ANSI standards;
b) approved and draft international and regional standards (e.g., ISO, IEC); and
c) approved and draft foreign standards (e.g., JIS and DIN).

For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

Additional availability contact information is provided below as needed.

Table 1 shows standards bodies and their web sites.

Table 1 — Standards bodies

Abbreviation	Standards body	Web site
ANSI	American National Standards Institute	http://www.ansi.org
IEC	International Engineering Consortium	http://www.iec.ch
INCITS	International Committee for Information Technology Standards	http://www.incits.org
ISO	International Standards Organization	http://www.iso.ch
T10	INCITS T10 SCSI storage interfaces	http://www.t10.org

2.2 Approved references

At the time of publication, the following referenced standards or technical reports were approved:

ISO/IEC 14776-414, *SCSI Architecture Model-4 (SAM-4) (ANSI INCITS 447-2008)*

2.3 References under development

At the time of publication, the following referenced standards were still under development. For information on the current status of the document, or regarding availability, contact the relevant standards body or other organization as indicated.

ISO/IEC 14776-454, SCSI Primary Commands-4 (SPC-4) (T10/1731-D)

NOTE 1 — For more information on the current status of these documents, contact the INCITS Secretariat at 202-737-8888 (phone), 202-638-4922 (fax) or via Email at incits@itic.org. To obtain copies of these documents, contact Global Engineering at 15 Inverness Way, East Englewood, CO 80112-5704 at 303-792-2181 (phone), 800-854-7179 (phone), or 303-792-2192 (fax) or see http://www.incits.org.

2.4 Other references

For information on the current status of the listed documents, or regarding availability, contact the indicated organization.

Universal Serial Bus Specification Revision 2.0 (USB-2). April 27, 2000

Universal Serial Bus 3.0 Specification Revision 1.0 (USB-3). November 12, 2008

Universal Serial Bus Mass Storage Class Specification Overview Rev 1.3 (MSC). September 5, 2008

NOTE 2 — For information on the current status of USB documents, see the USB Implementers Forum at http://www.usb.org.

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 application client: An object that is the source of SCSI commands (see SAM-4).

3.1.2 Bulk-in Endpoint Descriptor: A USB Endpoint Descriptor with the BM ATTRIBUTES field set to 02h and bit 7 of the bEndPoint address field set to 1.

3.1.3 Bulk-in pipe: used to transfer data and status from the UAS target port to the UAS initiator port.

3.1.4 Bulk-out Endpoint Descriptor: A USB Endpoint Descriptor with the BM ATTRIBUTES field set to 02h and bit 7 of the bEndPoint address field set to 0.

3.1.5 Bulk-out pipe: used to transfer data and commands from the UAS initiator port to the UAS target port.

3.1.6 Default pipe: The message pipe created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero (see USB-2).

3.1.7 Information Unit (IU): An IU is a formatted collection of data that carries command, task management function, sense, reponse, read ready, or write ready information (see 6.2).

3.1.8 logical unit number (LUN): A 64-bit identifier for a logical unit (see SAM-4).

3.1.9 Pipe: A logical abstraction using USB endpoints representing an association between a function (see USB-2 and USB-3) of a USB device and an application client.

3.1.10 read data: Data transferred to the SCSI application client's data-in buffer from the SCSI device server, as requested by the Send Data-In transport protocol service.

3.1.11 Service Deliver Subsystem: The part of the USB I/O system which transmits information between the UAS initiator port and the UAS target port (e.g., USB hubs, USB cables).

3.1.12 task manager: An object that controls the sequencing of commands and processes task management functions. See SAM-4.

3.1.13 transaction packet (TP): A type of header packet used to communicate information between a UAS target device and a UAS initiator device (see USB-3).

3.1.14 UAS domain: One UAS initiator port and one or more UAS target ports (see 4.4).

3.1.15 UAS initiator device: A USB host (see USB-2 and USB-3) that contains one or more UAS initiator ports.

3.1.16 UAS initiator port: The USB host (see USB-2) and USB host port components (see USB-2 and USB-3).

3.1.17 UAS target device: A USB device that contains one or more UAS target ports that attach to a UAS initiator device.

3.1.18 UAS target port: A USB interface that contains two USB Bulk-in endpoints, two USB Bulk-out endpoints and the default USB control endpoint.

3.1.19 USB device: One or more USB interfaces and the default control endpoint (see USB-2 and USB-3).

3.1.20 USB endpoint: A collection of characteristics describing the USB device implementation of a pipe (see USB-2 and USB-3).

3.1.21 USB interface: The description of one or more USB endpoints (see USB-2 and USB-3).

3.1.22 USB Packet: A unit of data formated for transmission over Super Speed USB or High Speed USB.

3.1.23 write data: Data transferred from the SCSI application client's data-out buffer to the SCSI device server, as requested by the Request Data-Out transport protocol service.

3.2 Symbols and abbreviations

Abbreviation	Meaning
x	multiplication
/	division
≠ or NE	not equal
≤ or LE	less than or equal to
±	plus or minus
≈	approximately
+	add
-	subtract
< or LT	less than
= or EQ	equal
> or GT	greater than
≥ or GE	greater than or equal to
IU	Information Unit
LSB	Least significant bit
LUN	Logical unit number
MSB	Most significant bit
MSC	Mass Storage Class
SAM-4	SCSI Architecture Model-4
SCSI	Small Computer System Interface
SPC-4	SCSI Primary Commands-4
UAS	USB Attached SCSI (this standard)
USB	Universal Serial Bus (see USB-2 and USB-3)
USB-2	Universal Serial Bus Revision 2.0
USB-3	Universal Serial Bus 3.0 Revision 1.0

3.3 Keywords

3.3.1 invalid: A keyword used to describe an illegal or unsupported bit, byte, word, field or code value. Receipt by a device server of an invalid bit, byte, word, field or code value shall be reported as error.

3.3.2 mandatory: A keyword indicating an item that is required to be implemented as defined in this standard.

3.3.3 may: A keyword that indicates flexibility of choice with no implied preference.

3.3.4 may not: A keyword that indicates flexibility of choice with no implied preference.

3.3.5 obsolete: A keyword indicating that an item was defined in prior SCSI standards but has been removed from this standard.

3.3.6 option, optional: Keywords that describe features that are not required to be implemented by this standard. However, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

3.3.7 reserved: A keyword referring to bits, bytes, words, fields, and code values that are set aside for future standardization. A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words, or fields for zero values. Receipt of reserved code values in defined fields shall be reported as error.

3.3.8 shall: A keyword indicating a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

3.3.9 should: A keyword indicating flexibility of choice with a strongly preferred alternative.

3.3.10 vendor specific: Something (e.g., a bit, field, code value) that is not defined by this standard. Specification of the referenced item is determined by the SCSI device vendor and may be used differently in various implementations.

3.4 Editorial conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in subclase 3.1 or in the text where they first appear.

Upper case is used when referring to the name of a numeric value defined in this specification or a formal attribute possessed by an entity. When necessary for clarity, names of objects, procedure calls, arguments or discrete states are capitalized or set in bold type. Names of fields are identified using small capital letters (e.g., NACA bit).

Names of procedure calls are identified by a name in bold type (e.g., **Execute Command**). Names of arguments are denoted by capitalizing each word in the name (e.g., Sense Data is the name of an argument in the **Execute Command** procedure call).

Quantities having a defined numeric value are identified by large capital letters (e.g., CHECK CONDITION). Quantities having a discrete but unspecified value are identified using small capital letters. (e.g., TASK COMPLETE, indicates a quantity returned by the **Execute Command** procedure call). Such quantities are associated with an event or indication whose observable behavior or value is specific to a given implementation standard.

Lists sequenced by letters (e.g., a-red, b-blue, c-green) show no priority relationship between the listed items. Numbered lists (e.g., 1-red, 2-blue, 3-green) show a priority ordering between the listed items.

If a conflict arises between text, tables, or figures, the order of precedence to resolve the conflicts is text; then tables; and finally figures. Not all tables or figures are fully described in the text. Tables show data format and values.

Notes and examples do not constitute any requirements for implementors.

3.5 Numeric and character conventions

3.5.1 Numeric conventions

A binary number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B_FD8C_FA23h).

A decimal number is represented in this standard by any sequence of digits comprised of only the Arabic numerals 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

This standard uses the following conventions for representing decimal numbers:

a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
b) the thousands separator (i.e., separating groups of three digits in a portion of the number) is a space; and
c) the thousands separator is used in both the integer portion and the fraction portion of a number.

Table 2 shows some examples of decimal numbers using various numbering conventions.

Table 2 — Numbering conventions

French	English	This standard
0,6	0.6	0.6
3,141 592 65	3.14159265	3.141 592 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

A decimal number represented in this standard with an overline over one or more digits following the decimal point is a number where the overlined digits are infinitely repeating (e.g., $666.\overline{6}$ means 666.666 666... or 666 2/3, and $12.\overline{142\ 857}$ means 12.142 857 142 857... or 12 1/7).

3.5.2 Byte encoded character strings conventions

When this standard requires one or more bytes to contain specific encoded characters, the specific characters are enclosed in single quotation marks. The single quotation marks identify the start and end of the characters that are required to be encoded but are not themselves to be encoded. The characters that are to be encoded are shown in the case that is to be encoded.

An ASCII space character (i.e., 20h) may be represented in a string by the character '¬' (e.g., 'SCSI¬device').

The encoded characters and the single quotation marks that enclose them are preceded by text that specifies the character encoding methodology and the number of characters required to be encoded.

EXAMPLE - Using the notation described in this subclause, stating that eleven ASCII characters 'SCSI device' are to be encoded would be the same writing out the following sequence of byte values: 53h 43h 53h 49h 20h 64h 65h 76h 69h 63h 65h.

3.6 Sequence figure notation

A sequence figure describes sequences of communication between a requestor and a responder. Figure 2 is an example sequence figure. A line with an arrowhead that points to the responder represents a communication from the requestor to the responder. A line with an arrowhead that points to the requestor represents a communication from the responder to the requestor.

Each line with an arrowhead has a label. The label describes the communication between the requestor and the responder.

Communications that appear near the top of a sequence figure occur earlier in time than communications that appear below them.

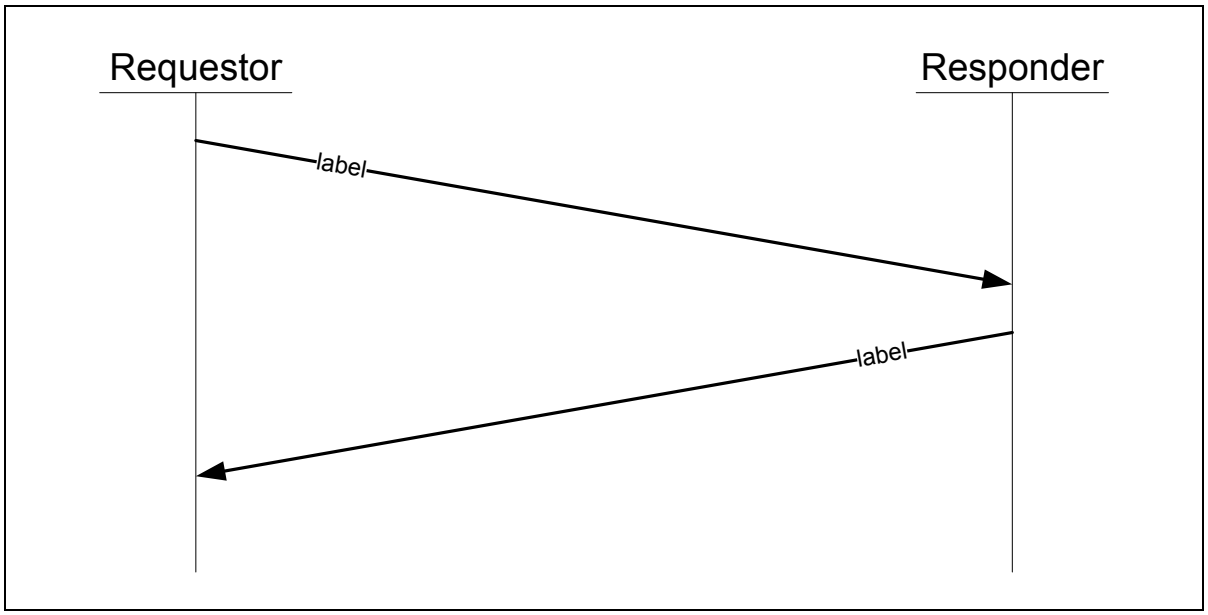


Figure 2 — Example Sequence figure

3.7 Notation for procedures and functions

In this standard, the model for functional interfaces between objects is the callable procedure. Such interfaces are specified using the following notation:

[Result =] Procedure Name (IN ([input-1] [,input-2] …), OUT ([output-1] [,output-2] …))

Where:

Result:A single value representing the outcome of the procedure or function.

Procedure Name:descriptive name for the function to be performed.

IN (Input-1, Input-2, …)A comma-separated list of names identifying caller-supplied input data objects.

OUT (Output-1, Output-2, …)A comma-separated list of names identifying output data objects to be returned by the procedure.

[…] Brackets enclose optional or conditional parameters and arguments.

This notation allows data objects to be specified as inputs and outputs.

In this standard, the notation Procedure Name () is used to indicate the name of a procedure without specifying the input data objects or output data objects.

4 Model

4.1 Overview

A UAS target port shall support a single I_T nexus. The minimum configuration for a UAS target port (see figure 3) consists of:

a) the Default pipe (see USB-2);
b) two Bulk-in pipes:
   A) Status pipe; and
   B) Data-in pipe;

   and

c) two Bulk-out pipes:
   A) Command pipe; and
   B) Data-out pipe.

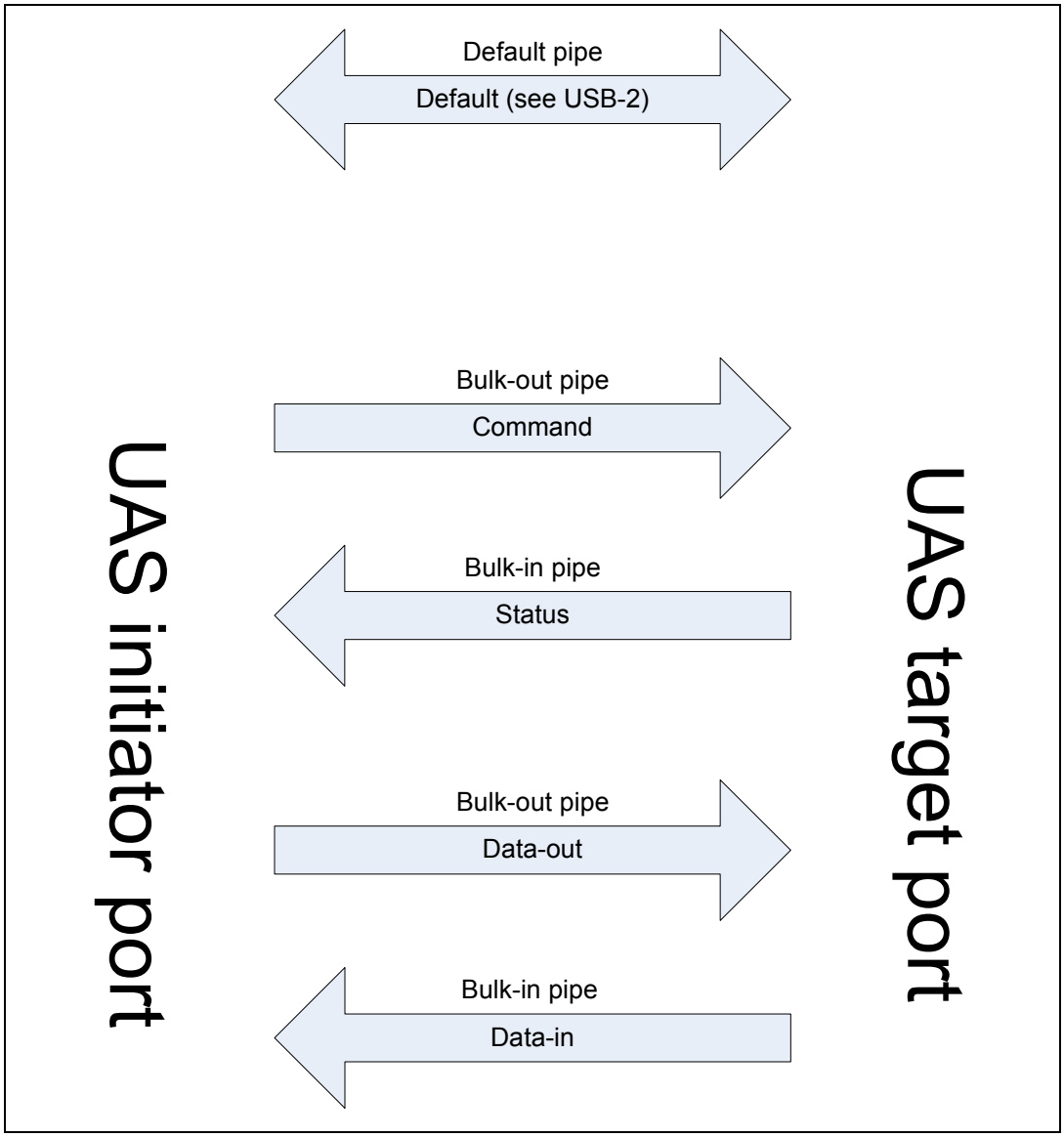


Figure 3 — USB Model

The Default pipe, required by both USB-2 and USB-3, is not defined in this standard.

The UAS target port receives IUs from the UAS initiator port using the Command pipe and responds with IUs using the Status pipe.

The Data-in pipe transmits read data (i.e., data to the application client's data-in buffer). The Data-out pipe transmits write data (i.e., data from the application client's data-out buffer).

The UAS target port shall have sufficient buffering or other resources available to receive commands after the USB device has entered the USB configured state (see USB-2 or USB-3). The UAS initiator port should have sufficient buffering available to receive status from the UAS target port after the USB device has entered the USB configured state (see USB-2 and USB-3).

If the UAS target port is unable to send status to the UAS initiator port when said status is available, then the target port may abort all commands in the task set and all commands that the target port receives until the UAS target port is able to terminate a command with CHECK CONDITION status with the sense key set to UNIT ATTENTION with the additional sense code set to COMMANDS CLEARED BY DEVICE SERVER. If:

a) the CREDIT_HP_TIMER (see USB-3) has been started and has not expired;
b) the CREDIT_HP_TIMER (see USB-3) has not been started and less than 5 ms has elapsed since the last transmission of an ACK packet from the UAS initiator port; or
c) less than 5 ms has elapsed since the last transmission of an IN packet (see USB-2) from the UAS initiator port for a High Speed device,

then the UAS target port shall not take any action to abort commands resulting from a failure to be able to send status to the UAS initiator.

4.2 Tag handling

The TAG field in a COMMAND IU (see 6.2.2) contains the command identifier as defined in SAM-4. The TAG field in a TASK MANAGEMENT IU (see 6.2.7) is an association between a SAM-4 Received Task Management Function Executed and a SAM-4 Send Task Management Request. The number space used in the TAG fields is shared across COMMAND IUs and TASK MANAGEMENT IUs (i.e., if the same tag is used for a concurrent COMMAND IU and TASK MANAGEMENT IU, then the device shall report an error as defined in this subclause).

If a UAS target device performs tag checking and a UAS target port calls SCSI Command Received () with a tag already in use by another command (i.e., an overlapped command) in any logical unit, then the task router and task manager(s) shall:

a) abort all task management functions received on that I_T nexus; and
b) respond to the overlapped command as defined in SAM-4.

If a UAS target device performs tag checking and:

a) a UAS target port calls SCSI Command Received () with a tag already in use by a task management function in any logical unit; or
b) a UAS target port calls Task Management Request Received () with a tag already in use by a command or task management function in any logical unit,

then the task router and task manager(s) shall:

a) abort all commands received on that I_T nexus;
b) abort all task management functions received on that I_T nexus; and
c) call Task Management Function Executed () with the Service Response set to FUNCTION REJECTED - Overlapped Tag Attempted (i.e., requesting that the target port set the RESPONSE CODE field set to OVERLAPPED TAG ATTEMPTED).

4.3 Data transfers

The UAS model described in 4.1 enables a UAS target port to process commands and return status at the same time that data is being transferred for other commands. The UAS target port should be able to perform the following concurrently:

a) transfer data;
b) accept COMMAND IUs and TASK MANAGEMENT IUs using the Command pipe;

c) process the commands and task management functions; and
d) return status for commands and task management functions on the Status pipe.

See 6.3.8 for an example of the concurrent command operation and task management operation described in this subclause.

If the task set is full and the UAS target port receives a command, then the UAS target port shall return a SENSE IU using the Status pipe with a status of TASK SET FULL. The SENSE IU is returned on the Status pipe and may be returned while data is transferred on the Data-out pipe or Data-in pipe for a different command.

If the UAS target device returns a READ READY IU or a WRITE READY IU on the Status pipe, then the UAS target device shall be ready to send or receive all the data for the indicated I_T_L_Q nexus. After the last byte of data is transferred and achieved a USB acknowledgment (see 7.16), the UAS target device shall return a SENSE IU on the Status pipe to indicate command completion. After the command is complete, the associated Data-out pipe or Data-in pipe may be used to transfer data for a different command.

UAS target ports that connect via SuperSpeed (see USB-3) shall send an ERDY transaction packet on the Status pipe using the tag from the COMMAND IU as the stream ID when the UAS target port has a STATUS IU or a RESPONSE IU to transmit.

4.4 UAS domain

Figure 4 shows an example of a simple UAS domain that contains:

a) one UAS initiator; and
b) one USB device.

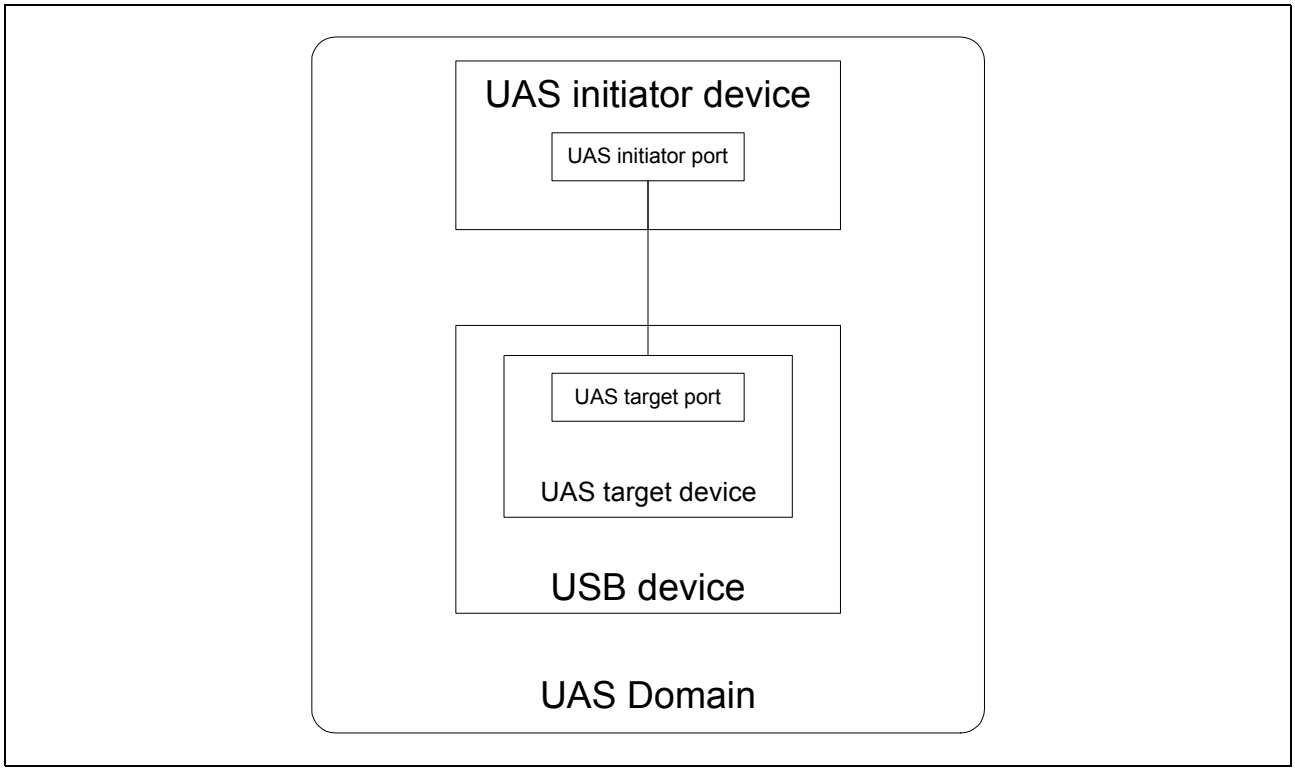


Figure 4 — Example Simple UAS domain

Figure 5 shows an example of a UAS domain that contains:

a) one UAS initiator; and
b) several USB devices.

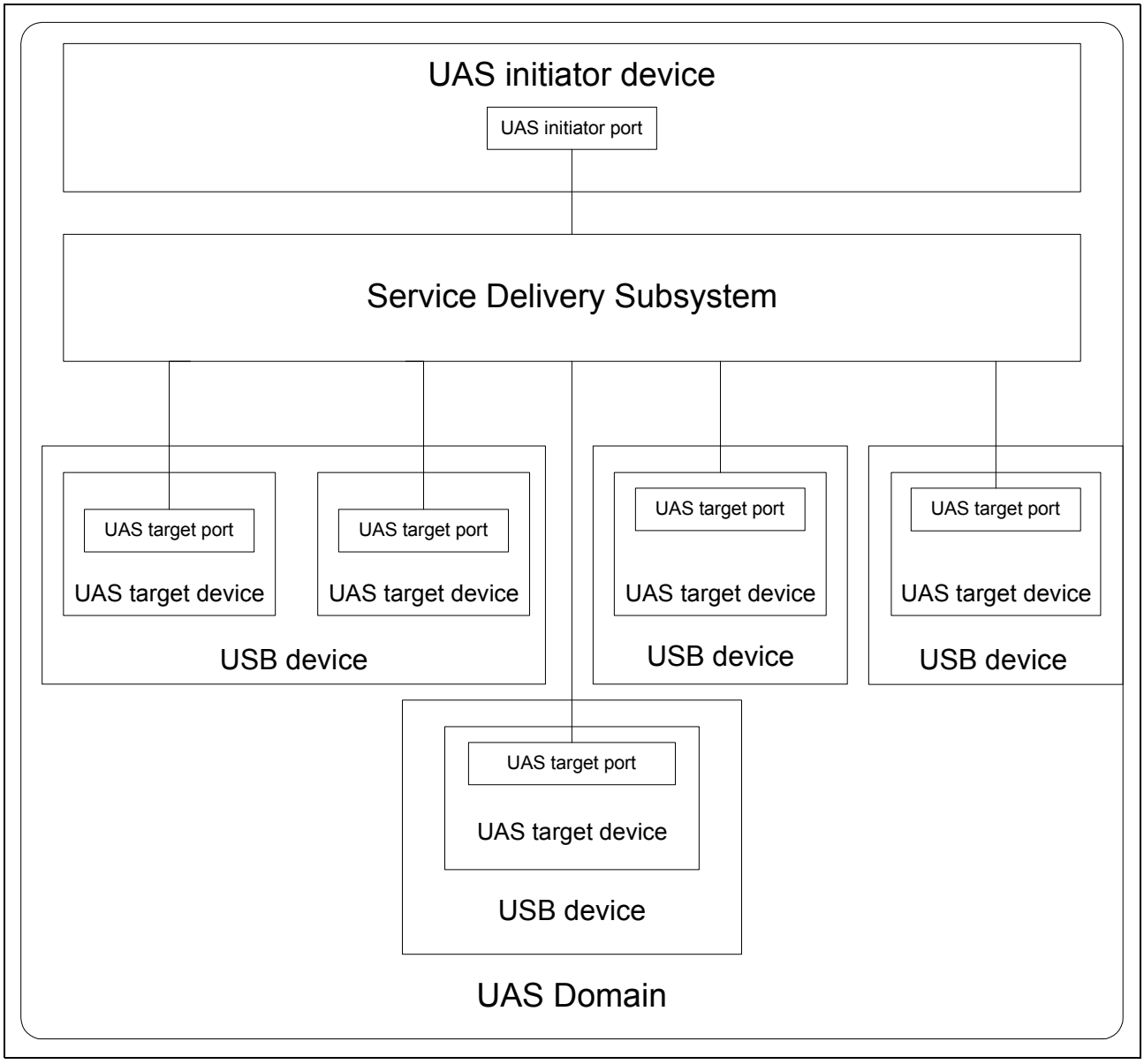


Figure 5 — Example Complex UAS Domain

4.5 Addressing

The SAM-4 port identifier is the USB address (see USB-2) or the USB route string (see USB-3) assigned as a part of the USB enumeration process.

There is only one initiator port in a UAS domain. The UAS initiator port identifier shall be set to one. This standard does not make use of the UAS initiator port identifier.

4.6 World wide name

The Device Identification VPD page (see SPC-4) shall contain at least one designation descriptor with the DESIGNATOR TYPE field set to 03h (i.e., NAA) and the ASSOCIATION field set to 00b (i.e., logical unit) for each logical unit accessible through that UAS target port.

4.7 Resets

A UAS target device shall perform the operations for all reset conditions resulting from SCSI events as defined in SAM-4 with the additions in this subclause.

A USB bus reset (see USB-2 or USB-3) shall be treated as a hard reset event (see SAM-4).

4.8 I_T Nexus loss

If the UAS target port and UAS initiator port are disconnected then the device shall perform the actions for I_T Nexus Loss as defined in SAM-4. A UAS target port is disconnected from the UAS initiator port when loss of signal is detected or the UAS initiator port fails to respond within the USB-3 or USB-2 timeouts.

4.9 Target power loss expected

If the UAS target device detects that it may lose power (e.g., a battery is running low on power), then the USB target port should establish a unit attention condition with additional sense code set to WARNING - POWER LOSS EXPECTED.

4.10 USB error handling

In USB-2 the W MAX PACKET SIZE field is 512 bytes and in USB-3 the W MAX PACKET SIZE field is 1 024 bytes (see 5.2.3.4). Communication on any pipe may consist of short packets (i.e., packets that are less than the contents of the W MAX PACKET SIZE field). All packets on the Status pipe and Command pipe may be short packets, this is not an error.

No condition defined in this standard results in a stall (see USB-2) on any pipe.

5 USB

5.1 Overview

This clause describes information associated with USB to support this standard (e.g., USB descriptors).

5.2 USB resource requirements

5.2.1 Overview

This standard requires a minimum of:

a) one Device Descriptor (see 5.2.3.1);
b) one Configuration Descriptor (see 5.2.3.2);
c) one Interface Descriptor (see 5.2.3.3); and
d) four Endpoint Descriptors (see 5.2.3.4).

The USB Get Descriptor request (see USB-2 and USB-3) returns the descriptors defined in 5.2.3.

5.2.2 USB class specific requests

There are no USB Class Specific Requests defined in this standard.

5.2.3 USB descriptors

5.2.3.1 Device descriptor

Table 3 describes the Device descriptor format.

Table 3 — Device descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	B LENGTH (12h)							
1	B DESCRIPTOR TYPE (01h)							
2	BCD USB							(LSB)
3	(MSB)							
4	B DEVICE CLASS (00h)							
5	B DEVICE SUBCLASS (00h)							
6	B DEVICE PROTOCOL (00h)							
7	B MAX PACKET SIZE							
8	ID VENDOR							(LSB)
9	(MSB)							
10	ID PRODUCT							(LSB)
11	(MSB)							
12	BCD DEVICE							(LSB)
13	(MSB)							
14	I MANUFACTURER							
15	I PRODUCT							
16	I SERIAL NUMBER							
17	B NUM CONFIGURATIONS							

The:

a) B LENGTH field;
b) B DESCRIPTOR TYPE field;

c) B DEVICE CLASS field;
d) B DEVICE SUBCLASS field; and
e) B DEVICE PROTOCOL field,

shall be set to the value defined in table 3 (see USB-2 and USB-3).

See USB-2 and USB-3 for the description of the:

a) BCD USB field;
b) B MAX PACKET SIZE field;
c) ID VENDOR field;
d) ID PRODUCT field;
e) BCD DEVICE field;
f) I MANUFACTURE field;
g) I PRODUCT field;
h) I SERIAL NUMBER field; and
i) B NUM CONFIGURATIONS field.

5.2.3.2 Configuration descriptor

Table 4 describes the Configuration descriptor format.

Table 4 — Configuration descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	B LENGTH (09h)							
1	B DESCRIPTOR TYPE (02h)							
2	W TOTAL LENGTH							(LSB)
3	(MSB)							
4	B NUM INTERFACES							
5	B CONFIGURATION VALUE							
6	I CONFIGURATION							
7	Reserved	SELF POWERED	REMOTE WAKEUP	Reserved				
8	MAX POWER							

The:

a) B LENGTH field; and
b) B DESCRIPTOR TYPE field,

shall be set to the value defined in table 4 (see USB-2 and USB-3).

See USB-2 and USB-3 for the description of the:

a) W TOTAL LENGTH field;
b) B NUM INTERFACES field;
c) B CONFIGURATION VALUE field;
d) I CONFIGURATION field;
e) SELF POWERED field;
f) REMOTE WAKEUP field; and
g) MAX POWER field.

5.2.3.3 Interface descriptor

Table 5 describes the Interface descriptor format.

Table 5 — Interface Descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	B LENGTH (09h)							
1	B DESCRIPTOR TYPE (04h)							
2	B INTERFACE NUMBER							
3	B ALTERNATE SETTING							
4	B NUM ENDPOINTS							
5	B INTERFACE CLASS (08h)							
6	B INTERFACE SUBCLASS (06h)							
7	B INTERFACE PROTOCOL (62h)							
8	I INTERFACE							

The:

a) B LENGTH field;
b) b descriptor type field;
c) B INTERFACE CLASS field;
d) B INTERFACE SUBCLASS field; and
e) B INTERFACE PROTOCOL field,

shall be set to the value defined in table 5 (see USB-2, USB-3, and MSC).

See USB-2 and USB-3 for the description of the:

a) B INTERFACE NUMBER field;
b) B ALTERNATE SETTING field;
c) B NUM ENDPOINTS field; and
d) I INTERFACE field.

5.2.3.4 Endpoint descriptors

Table 6 describes the Bulk-in endpoint descriptor descriptor format.

Table 6 — Bulk-in endpoint descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	B LENGTH (07h)							
1	B DESCRIPTOR TYPE (05h)							
2	DIR (1b)	Reserved			ENDPOINT NUMBER			
3	BM ATTRIBUTES (02h)							
4	W MAX PACKET SIZE							(LSB)
5	(MSB)							
6	Reserved							

The:

a) B LENGTH field;
b) B DESCRIPTOR TYPE field;
c) DIR field; and

d) BM ATTRIBUTES field,

shall be set to the value defined in table 6 (see USB-2 and USB-3).

See USB-2 and USB-3 for the description of the ENDPOINT NUMBER field.

The W MAX PACKET SIZE field shall be set to 512 bytes for high speed devices (see USB-2). The W MAX PACKET SIZE field shall be set to 1 024 bytes for Super Speed devices (see USB-3).

Table 7 describes the Bulk-out endpoint descriptor format.

Table 7 — Bulk-out endpoint descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	B LENGTH (07h)							
1	B DESCRIPTOR TYPE (05h)							
2	DIR (0b)	Reserved			ENDPOINT NUMBER			
3	BM ATTRIBUTES (02h)							
4	W MAX PACKET SIZE							(LSB)
5	(MSB)							
6	Reserved							

The:

a) B LENGTH field;
b) B DESCRIPTOR TYPE field;
c) DIR field; and
d) BM ATTRIBUTES field,

shall be set to the value defined in table 6 (see USB-2 and USB-3).

See USB-2 and USB-3 for the description of the:

a) ENDPOINT NUMBER field; and
b) W MAX PACKET SIZE field.

5.3.3.5 Pipe Usage class specific descriptor

A Pipe Usage class specific descriptor shall be the first descriptor following each endpoint descriptor referenced by the Interface descriptor (see 5.2.3.3). Table 8 describes the format of the Pipe Usage Class specific descriptor.

Table 8 — Pipe Usage Descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	B LENGTH (04h)							
1	B DESCRIPTOR TYPE (24h)							
2	B PIPE ID							
3	Reserved							

The B LENGTH field shall be set to the value defined in table 8.

The B DESCRIPTOR TYPE field shall be set to the value defined in table 8 (see MSC).

The B PIPE ID field identifies the pipe associated with the endpoint descriptor (see table 9).

Table 9 — Pipe ID

Value	Description
00h	Reserved
01h	Command pipe
02h	Status pipe
03h	Data-in pipe
04h	Data-out pipe
05h-DFh	Reserved
E0h-EFh	Vendor specific
F0h-FFh	Reserved

6 Transport

6.1 Overview

This clause describes the transport protocol. This includes IUs, data transfer sequences, and transport management.

6.2 IUs

6.2.1 Overview

Table 10 is a summary of the Information Units (IUs) and the assoicated IU ID field.

Table 10 — IU ID field summary

IU ID	Description	Reference
00h	Reserved	
01h	COMMAND IU	6.2.2
02h	Reserved	
03h	SENSE IU	6.2.5
04h	RESPONSE IU	6.2.6
05h	TASK MANAGEMENT IU	6.2.7
06h	READ READY IU	6.2.3
07h	WRITE READY IU	6.2.4
08h..FFh	Reserved	

All IUs include the header defined in table 11 as the first bytes of the IU.

Table 11 — IU Header

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID							
1	Reserved							
2	(MSB)	TAG						
3								(LSB)

The IU ID field identifies the type of IU (see table 10).

The TAG field is described in 4.2.

If a UAS target port processes an IU with an IU ID field conaining a reserved value, then the UAS target port shall return a RESPONSE IU with the RESPONSE CODE field set to INVALID INFORMATION UNIT.

6.2.2 COMMAND IU

The COMMAND IU shall be contained in a single USB packet and shall not share a USB packet with any other IU. Table 12 defines the COMMAND IU.

TABLE 12 — COMMAND IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID (01h)							
1	Reserved							
2	(MSB)			TAG				
3								(LSB)
4	Reserved		COMMAND PRIORITY			TASK ATTRIBUTE		
5	Reserved							
6	ADDITIONAL CDB LENGTH (n dwords)						Reserved	
7	Reserved							
8	(MSB)			LOGICAL UNIT NUMBER				
15								(LSB)
16				CDB				
31								
32				ADDITIONAL CDB BYTES				
31+nx4								

The IU ID field shall be set to the value defined in table 12.

The TAG field is described in 4.2.

The COMMAND PRIORITY field specifies the relative scheduling of this command as defined in SAM-4.

Table 13 defines the TASK ATTRIBUTE field.

Table 13 — TASK ATTRIBUTE **field**

Code	Task Attribute	Description
000b	SIMPLE	Specifies that the command be managed according to the rules for a simple task attribute (see SAM-4).
001b	HEAD OF QUEUE	Specifies that the command be managed according to the rules for a head of queue task attribute (see SAM-4).
010b	ORDERED	Specifies that the command be managed according to the rules for an ordered task attribute (see SAM-4).
011b	Reserved	
100b	ACA	Specifies that the command be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-4).
101b..111b	Reserved	

The ADDITIONAL CDB LENGTH field contains the length in dwords (i.e., four bytes) of the ADDITIONAL CDB BYTES field.

The LOGICAL UNIT NUMBER field specifies the identifier of the logical unit. The structure of the LOGICAL UNIT NUMBER field shall be as defined in SAM-4. If the addressed logical unit does not exist, the task manager shall follow the rules for selection of incorrect logical units defined in SAM-4.

The CDB and ADDITIONAL CDB BYTES fields together contain the CDB to be interpreted by the addressed logical unit. Any bytes after the end of the actual CDB within the two fields shall be ignored (e.g., a six-byte CDB

occupies the first six bytes of the CDB field, the remaining ten bytes of the CDB field are ignored, and the ADDITIONAL CDB BYTES field is not present).

The contents of the CDB are defined in the SCSI command standards (e.g., SPC-4).

6.2.3 READ READY IU

The READ READY IU is sent by a UAS target port to inform the UAS initiator port that the UAS target port is ready to send data for a data-in command (see 6.3.5) or a bi-directional command (see 6.3.7). UAS target devices that connect via SuperSpeed (see USB-3) shall return an ERDY transaction packet (see USB-3) on the Data-in pipe using the tag as the stream ID instead of the READ READY IU on the Status pipe. The READ READY IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 14 describes the READ READY IU.

TABLE 14 — READ READY IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID (06h)							
1	Reserved							
2	(MSB)	TAG						
3								(LSB)

The IU ID field shall be set to the value defined in table 14.

The TAG field (see 4.2) shall be set to the tag of the command to which the IU pertains.

6.2.4 WRITE READY IU

The WRITE READY IU is sent by a UAS target port to request write data from the UAS initiator port during a data-out command (see 6.3.4) or a bi-directional command (see 6.3.7). UAS target devices that connect via SuperSpeed (see USB-3) shall return an ERDY transaction packet (see USB-3) on the Data-out pipe using the tag as the stream ID instead of the WRITE READY IU on the Status pipe. The WRITE READY IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 15 defines the WRITE READY IU.

TABLE 15 — WRITE READY IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID (07h)							
1	Reserved							
2	(MSB)	TAG						
3								(LSB)

The IU ID field shall be set to the value defined in table 15.

The TAG field (see 4.2) shall be set to the tag of the command to which the IU pertains.

6.2.5 SENSE IU

The SENSE IU is sent by the UAS target port to deliver SCSI status. The SENSE IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 16 defines the SENSE IU.

Table 16 — SENSE IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID (03h)							
1	Reserved							
2	(MSB)			TAG				
3								(LSB)
4	STATUS QUALIFIER							
5								
6	STATUS							
7..13	Reserved							
14				LENGTH (n-15)				
15								
16	(MSB)			SENSE DATA				
n								(LSB)

The IU ID field shall be set to the value defined in table 16.

The TAG field (see 4.2) shall be set to the tag of the command to which the IU pertains.

The STATUS QUALIFIER field shall be set to the status qualifier for the command (see SAM-4);

The STATUS field shall be set to the status code (see SAM-4) for the command that has completed.

TThe LENGTH field contains the number of bytes that follow in the SENSE IU. If no sense data is available, then the LENGTH field shall be set to 0000h.

The SENSE DATA field shall be set to the sense data, if any, for the command associated with the tag (see SAM-4).

6.2.6 RESPONSE IU

The RESPONSE IU is used to pass task management status information from the UAS target port to the UAS initiator port. The RESPONSE IU may be returned in response to a COMMAND IU as a means to report an error condition detected by the transport. Each RESPONSE IU shall be sent in a single USB packet. The RESPONSE IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 17 defines the RESPONSE IU.

Table 17 — RESPONSE IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID (04h)							
1	Reserved							
2	(MSB)			TAG				
3								(LSB)
4				ADDITIONAL RESPONSE INFORMATION				
6								
7	RESPONSE CODE							

The IU ID field shall be set to the value defined in table 17.

The TAG field (see 4.2) shall be set to the tag of the command to which the IU pertains.

The ADDITIONAL RESPONSE INFORMATION field contains additional response information for certain task management functions (e.g., QUERY ASYNCHRONOUS EVENT) as defined in SAM-4. ADDITIONAL RESPONSE INFORMATION shall be set to zero if the task management function does not define ADDITIONAL RESPONSE INFORMATION or the logical unit does not support response information.

The RESPONSE CODE field (see table 18) indicates the status of a task management function.

Table 18 — RESPONSE CODE **field**

Code	Description	Task [b]	Command [c]
00h	TASK MANAGEMENT FUNCTION COMPLETE	Valid	Invalid
01h	Reserved	Invalid	Invalid
02h	INVALID INFORMATION UNIT	Valid	Valid
03h	Reserved	Invalid	Invalid
04h	TASK MANAGEMENT FUNCTION NOT SUPPORTED	Valid	Invalid
05h	TASK MANAGEMENT FUNCTION FAILED	Valid	Invalid
06h..07h	Reserved	Invalid	Invalid
08h	TASK MANAGEMENT FUNCTION SUCCEEDED	Valid	Invalid
09h	INCORRECT LOGICAL UNIT NUMBER	Valid	Invalid
0Ah	OVERLAPPED TAG ATTEMPTED [a]	Valid	Valid
0Bh-FFh	Reserved	Invalid	Invalid

[a] Returned in case of command/task management function or task management function/task management function tag conflicts.

[b] The Task column indicates the valid and invalid response codes returned by the UAS target device in response to a TASK MANAGEMENT IU.

[c] The Command column indicates the valid and invalid response codes returned by the UAS target device in response to a COMMAND IU.

6.2.7 TASK MANAGEMENT IU

The TASK MANAGEMENT IU is sent by a UAS initiator port to request that a task management function be processed by the task manager in a logical unit. The TASK MANAGEMENT IU shall be contained in a single

USB packet and shall not share a USB packet with any other Information Unit. Table 19 defines the TASK MANAGEMENT IU format.

Table 19 — TASK MANAGEMENT IU

Bit / Byte	7	6	5	4	3	2	1	0
0	IU ID (05h)							
1	Reserved							
2	(MSB)	TAG						
3								(LSB)
4	TASK MANAGEMENT FUNCTION							
5	Reserved							
6	(MSB)	TAG OF TASK TO BE MANAGED						
7								(LSB)
8	(MSB)	LOGICAL UNIT NUMBER						
15								(LSB)

The IU ID field shall be set to the value defined in table 19.

The TAG field is described in 4.2.

Table 20 defines the TASK MANAGEMENT FUNCTION field.

If the TASK MANAGEMENT FUNCTION field is set to 01h (i.e., ABORT TASK) or 80h (i.e., QUERY TASK) , then the TAG OF TASK TO BE MANAGED field specifies the tag from the COMMAND IU that contained the command to be aborted or queried. For all other task management functions, the TAG OF TASK TO BE MANAGED field is reserved.

Table 20 — TASK MANAGEMENT FUNCTION field

Code	Task management function	Uses the		Description
		LUN [a]	TOTTBM [b]	
00h	Reserved			
01	ABORT TASK	yes	yes	The task manager shall perform the ABORT TASK task management function using the value of the LOGICAL UNIT NUMBER field and the value of the TAG OF TASK TO BE MANAGED field to determine the task to be aborted (see SAM-4). [c]
02h	ABORT TASK SET	yes	no	The task manager shall perform the ABORT TASK SET task management function using the value of the LOGICAL UNIT NUMBER field to determine the task set to be aborted (see SAM-4). [c]
03h	Reserved			
04h	CLEAR TASK SET	yes	no	The task manager shall perform the CLEAR TASK SET task management function using the value of the LOGICAL UNIT NUMBER field to determing the task set to be cleared (see SAM-4). [c]
05h-07h	Reserved			
08h	LOGICAL UNIT RESET	yes	no	The task manager shall perform the LOGICAL UNIT RESET task management function using the value of the LOGICAL UNIT NUMBER field to determine the logical unit to be reset (see SAM-4). [c]
09h-0Fh	Reserved			
10h	I_T NEXUS RESET	no	no	The task manager shall perform the I_T NEXUS RESET task management function (see SAM-4). [c]
11h-3Fh	Reserved			
40h	CLEAR ACA	yes	no	The task manager shall perform the CLEAR ACA task management functionusing the value of the LOGICAL UNIT NUMBER field (see SAM-4). [c]
41h-7Fh	Reserved			
80h	QUERY TASK	yes	yes	The task manager shall perform the QUERY TASK task management function using the value of the LOGICAL UNIT NUMBER field and the value of the TAG OF TASK TO BE MANAGED field to determing the task to be queried (see SAM-4). [c]
81h	QUERY TASK SET	yes	no	The task manager shall perform the QUERY TASK SET task management function using the value of the LOGICAL UNIT NUMBER field to determine the task set to be queried (see SAM-4). [c]
82h	QUERY ASYNCHRONOUS EVENT	yes	no	The task manager shall perform the QUERY ASYNCHRONOUS EVENT task management function using the value of the LOGICAL UNIT NUMBER field (see SAM-4). [c]
83h-FFh	Reserved			

[a] LUN is the LOGICAL UNIT NUMBER field.
[b] TOTTBM is the TAG OF TASK TO BE MANAGED field.
[c] The task manager shall perform the specified task management function on the I_T nexus of the UAS initiator port and the UAS target port involved in the connection used to deliver the TASK MANAGEMENT IU.

If the TASK MANAGEMENT FUNCTION field contains a reserved or unsupported value, then the task manager shall return a RESPONSE IU with the RESPONSE CODE field set to TASK MANAGEMENT FUNCTION NOT SUPPORTED.

The LOGICAL UNIT NUMBER field contains the address of the logical unit. The structure of the LOGICAL UNIT NUMBER field shall be as defined in SAM-4. If the addressed logical unit does not exist, the task manager shall return a RESPONSE IU with the RESPONSE CODE field set to INCORRECT LOGICAL UNIT NUMBER.

6.3 Information unit sequences

6.3.1 Overview

The sequence figures in 6.3 describe communication between a UAS initiator port and a UAS target port. Figure 6 is an example UAS Sequence figure. Lines with an arrow that points to the UAS target port represents a communication from the UAS initiator port to the UAS target port. Lines with an arrow that points to the UAS initiator port represents a communication from the UAS target port to the UAS initiator port. Each arrow has the following:

a) a pipe name that is the name of a USB pipe (see 4.1);
b) an IU that is an optional parameter that indicates the IU (see 6.2) transferred on the pipe; and
c) a TAG_x that is an optional parameter that provides information regarding a field in an IU.

Only the Command pipe (see 4.1) and the Status pipe (see 4.1) use the IU and TAG_x parameters. The Data-in pipe (see 4.1) and Data-out pipe (see 4.1) transfer data associated with commands.

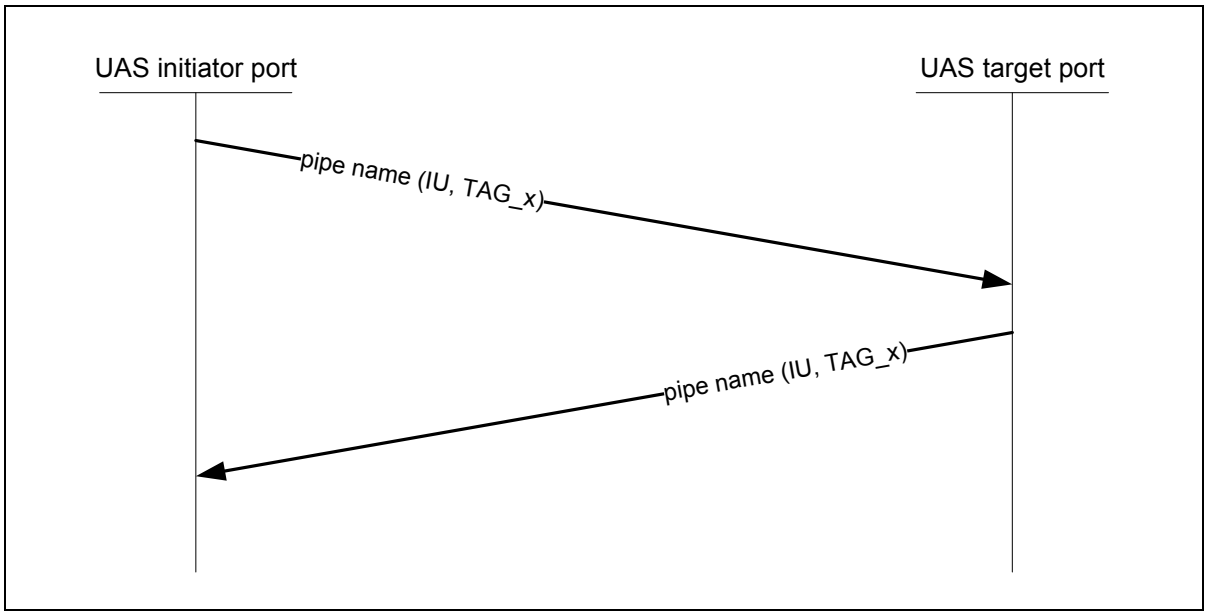


Figure 6 — UAS sequence figure notation

6.3.2 Non-data command/sense sequence

Figure 7 describes the sequence of communication between the UAS initiator port and UAS target port for a command that does not require data transfer.

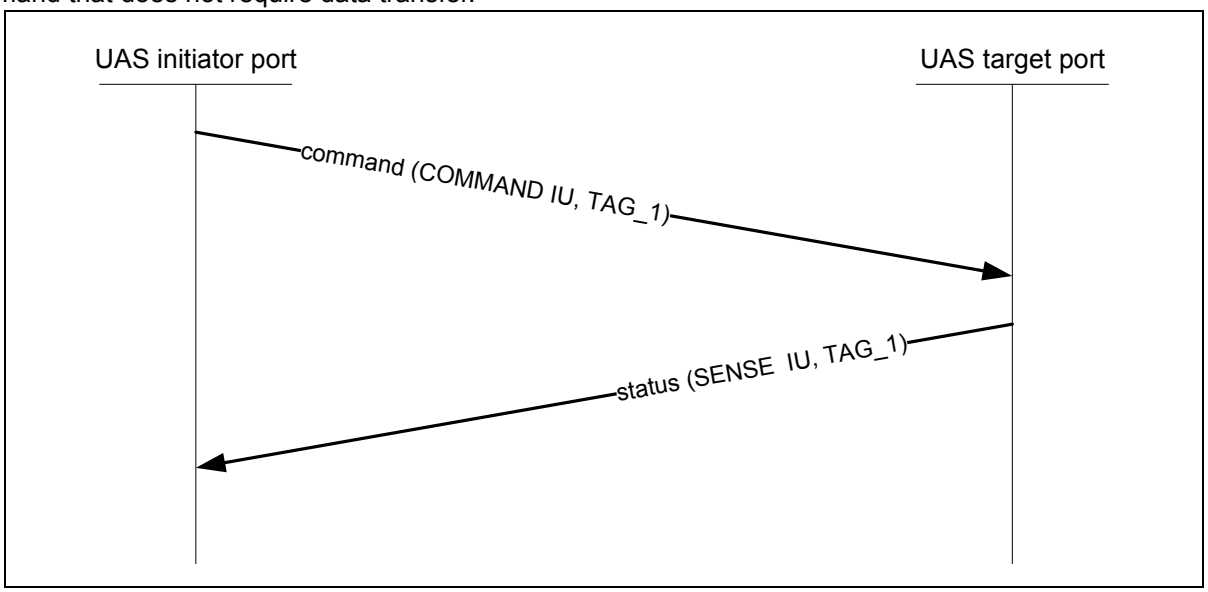


Figure 7 — Non-data transfer with Sense

6.3.3 Non-data command/response sequence

Figure 8 describes the sequence of communication between the UAS initiator port and UAS target port for a command that returns a RESPONSE IU (e.g., the UAS target port indicates an OVERLAPPED TAG ATTEMPTED).

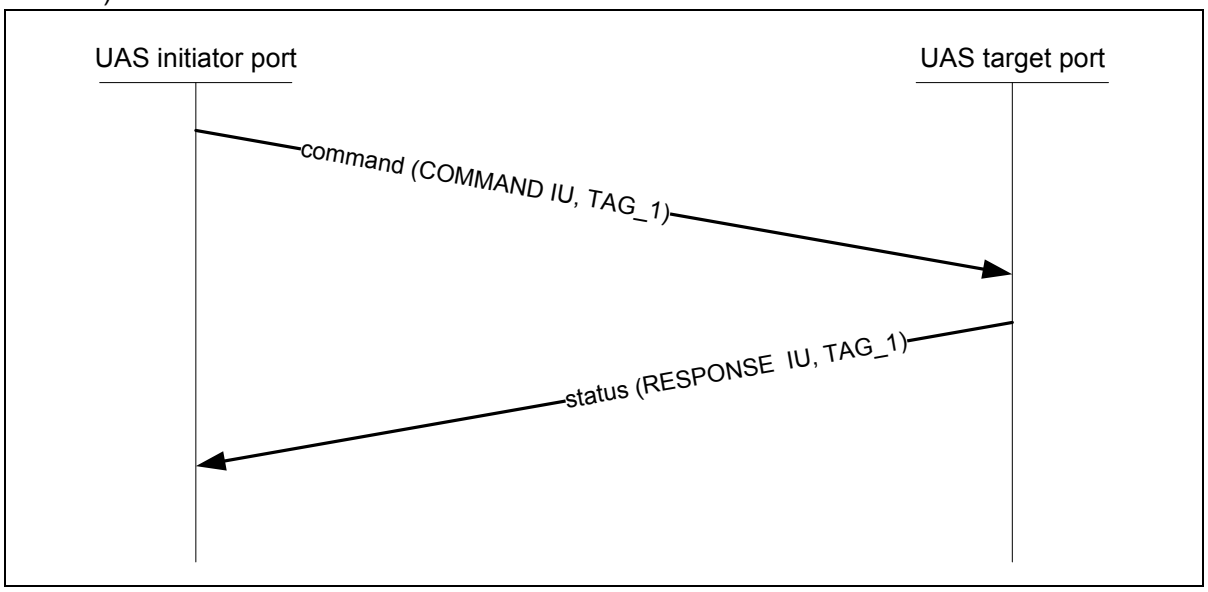
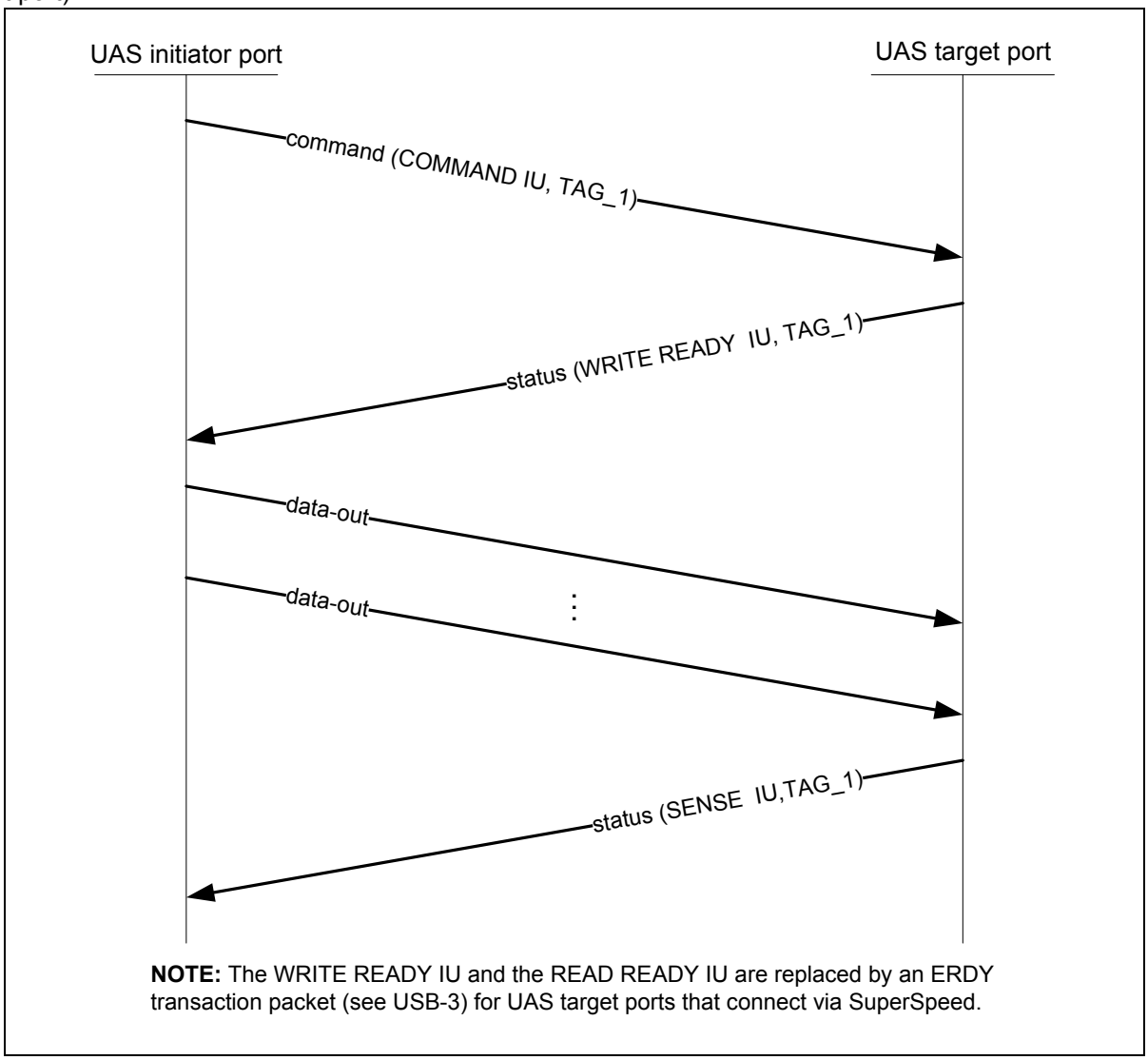


Figure 8 — Non-data Transfer with Response

6.3.4 Data-out command sequence

Figure 9 describes the sequence of communication between the UAS initiator port and UAS target port for a data-out command sequence (i.e., command that requires data transfer from the UAS initiator port to the UAS target port).

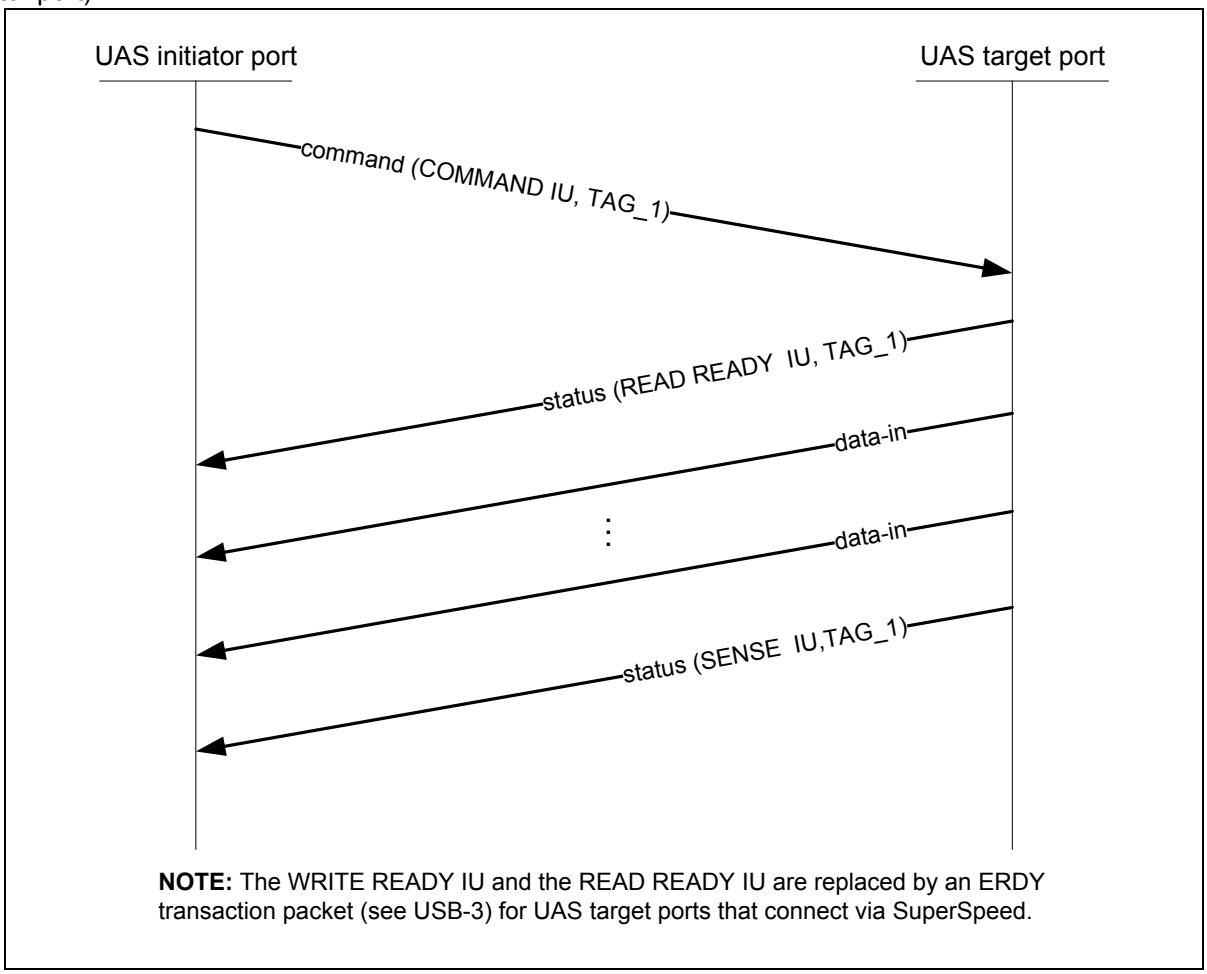


NOTE: The WRITE READY IU and the READ READY IU are replaced by an ERDY transaction packet (see USB-3) for UAS target ports that connect via SuperSpeed.

Figure 9 — Write Data Transfer

6.3.5 Data-in command sequence

Figure 10 describes the sequence of communication between the UAS initiator port and UAS target port for a Data-in command sequence (i.e., command that requires data transfer from the UAS target port to the UAS initiator port).



NOTE: The WRITE READY IU and the READ READY IU are replaced by an ERDY transaction packet (see USB-3) for UAS target ports that connect via SuperSpeed.

Figure 10 — Read Data Transfer

6.3.6 Task management function sequence

Figure 11 describes the sequence of communication between the UAS initiator port and UAS target port for a task management function.

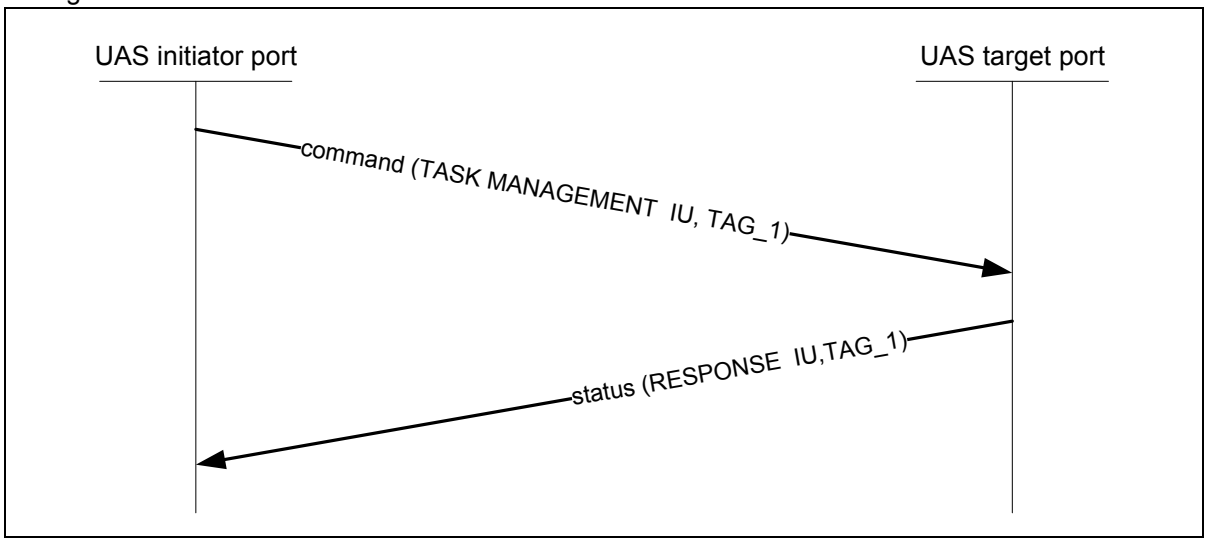
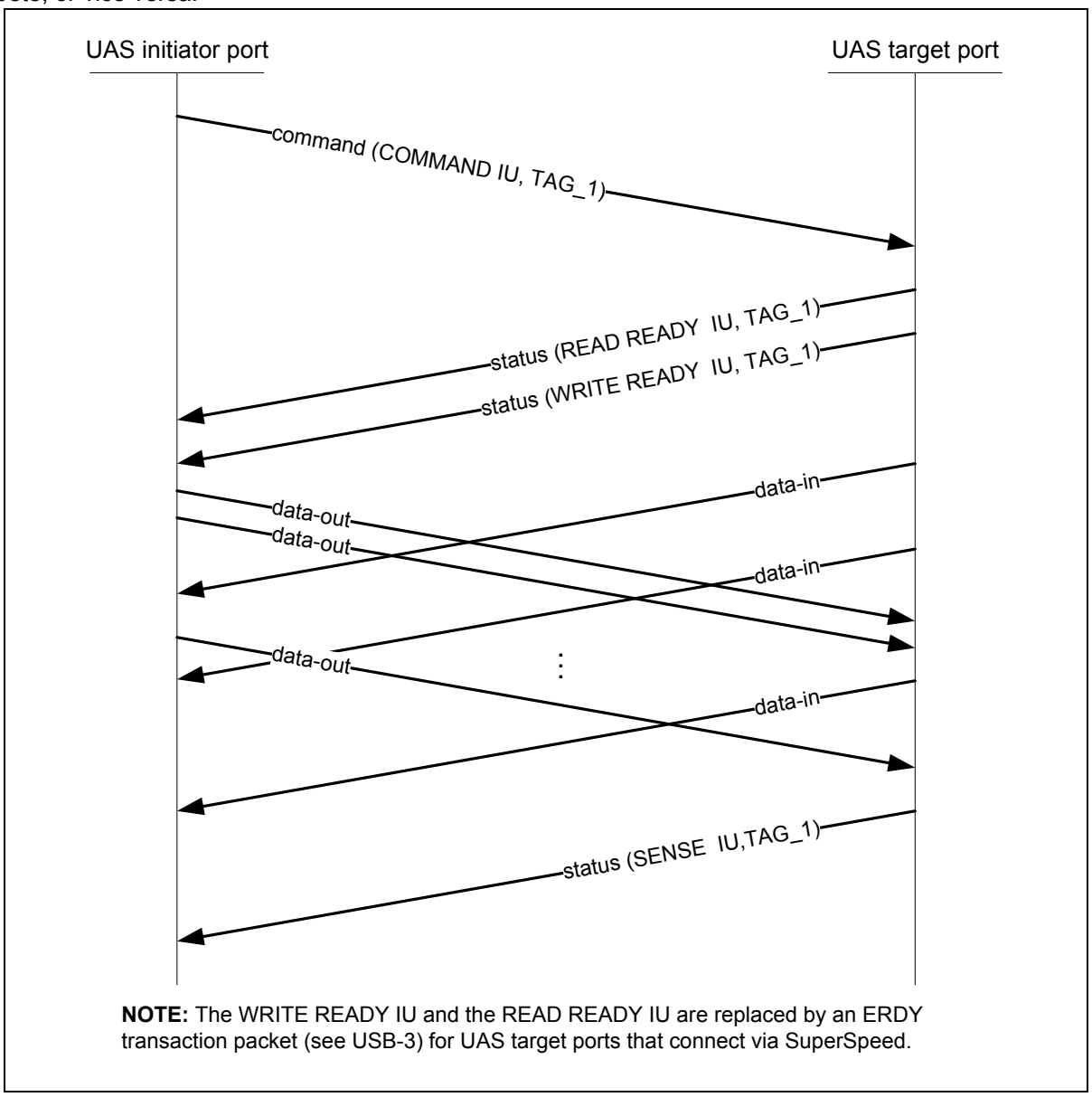


Figure 11 — Task Management

6.3.7 Bi-directional command sequence

Figure 12 describes the sequence of communication between the UAS initiator port and UAS target port for a command that requires data transfer both directions between the UAS target port and the UAS initiator port. Once the READ READY IU and WRITE READY IU are received by the UAS initiator port, both data-in and data-out transfer may occur asynchronously. The UAS target port may send the READ READY IU and wait for the data-in transfer to complete and then send the WRITE READY IU and then wait for data-out transfer to compete, or vice versa.



NOTE: The WRITE READY IU and the READ READY IU are replaced by an ERDY transaction packet (see USB-3) for UAS target ports that connect via SuperSpeed.

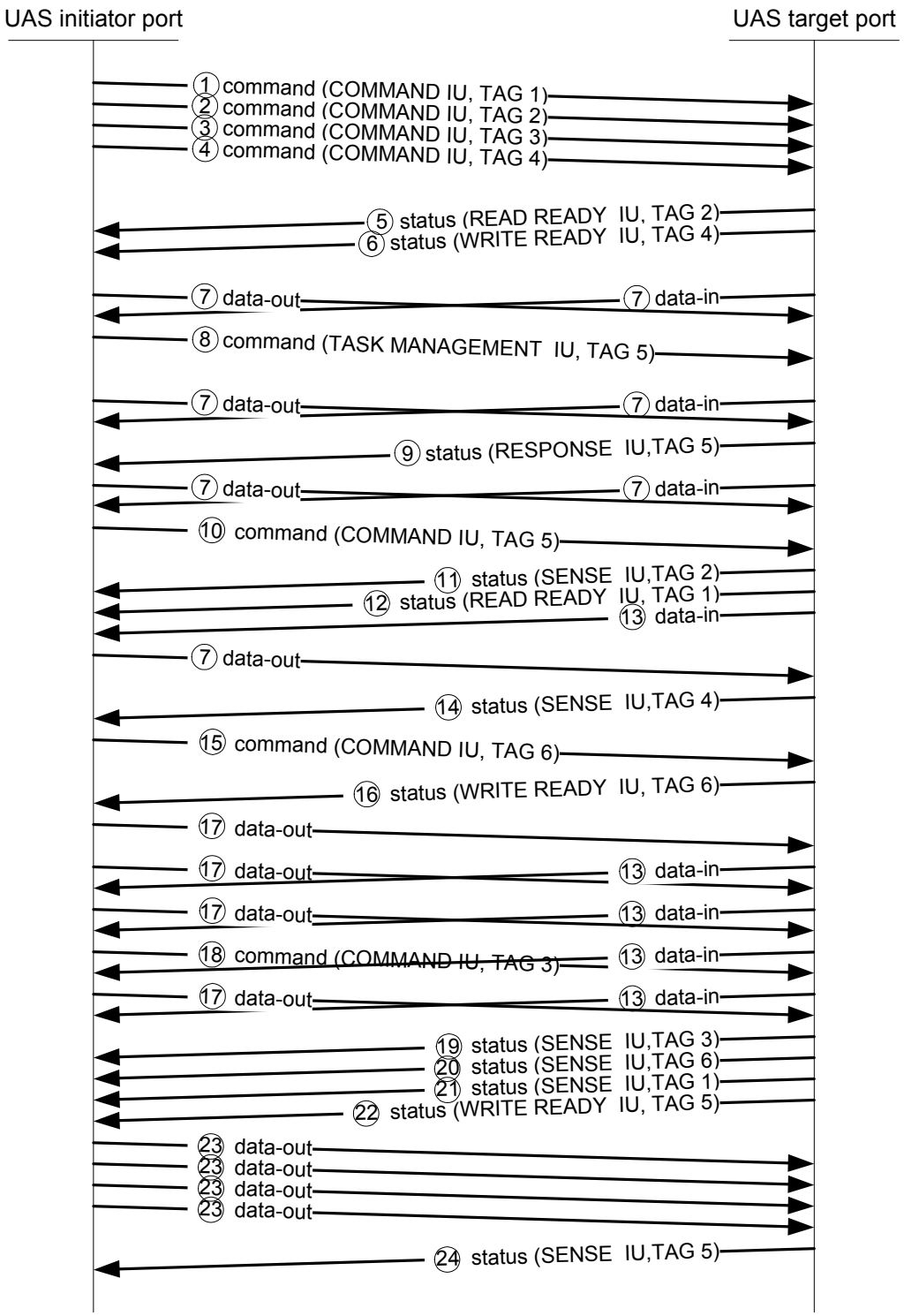
Figure 12 — Bi-directional Data Transfer

NOTE 3 — The order of the data-in and data-out phases of a bidirectional command may be influenced by both the definition of the bidirectional command and the capabilities of the USB target.

6.3.8 Multiple command example

Figure 13 describes the sequence of communication between a UAS initiator port and a UAS target port for several commands using task set management (see SAM-4) as follows:

1) the UAS initiator port transfers a read command with TAG 1;
2) the UAS initiator port transfers a read command with TAG 2;
3) the UAS initiator port transfers a write command with TAG 3;
4) the UAS initiator port transfers a write command with TAG 4;
5) the UAS target port requests to transfer the read data for TAG 2;
6) the UAS target port requests to transfer the write data for TAG 4;
7) data transfer begins for both TAG 2 and TAG 4;
8) the UAS initiator port transfers a task management request (using TAG 5) to abort the command with TAG 3;
9) the UAS target port reports that the command with TAG 3 was successfully aborted;
10) the UAS initiator port transfers a write command with TAG 5;
11) the UAS target port reports command completion for TAG 2;
12) the UAS target port requests to transfer the read data for TAG 1;
13) the UAS target port begins transferring data for TAG 1;
14) the UAS target port reports command completion for TAG 4;
15) the UAS initiator port transfers a write command with TAG 6;
16) the UAS target port requests the write data for TAG 6;
17) the UAS initiator port begins transferring data for TAG 6;
18) the UAS initiator port transfers a command that does not require data transfer with TAG 3;
19) the UAS target port reports command completion for TAG 3;
20) the UAS target port reports command completion for TAG 6;
21) the UAS target port reports command completion for TAG 1;
22) the UAS target port requests the write data for TAG 5;
23) the UAS initiator port begins transferring data for TAG 5;
24) the UAS target port reports command completion for TAG 5; and
25) the UAS target port is idle.



NOTE 1: Step numbers have been inserted in front of the pipe name to match the step numbers found in this subclause.

NOTE 2: The WRITE READY IU and the READ READY IU are replaced by an ERDY transaction packet (see USB-3) for UAS target ports that connect via SuperSpeed.

Figure 13 — Multiple Command Example

6.4 Transport requirements

After the UAS target port has started processing the Data-In delivery transport protocol service or the Data-Out delivery transport protocol service for the I_T_L_Q nexus the UAS target port shall complete the Data-In delivery transport protocol service or Data-Out delivery transport protocol service and terminate the command with a SENSE IU before beginning another Data-In delivery transport protocol service or Data-Out delivery transport protocol service for the same I_T nexus.

7 SCSI Application Layer Transport Protocol Services

7.1 SCSI transport protocol services overview

An application client requests the processing of a SCSI command by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following procedure call (see SAM-4):

Service response = Execute Command (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-in Buffer Size], [Data-out Buffer], [Data-out Buffer Size], [Command Priority]), OUT ([Data-in Buffer], [Sense Data], [Sense Data Length], Status, [Status Qualifier]))

This standard defines the transport protocol services required by SAM-4 in support of these procedure calls. Table 21 describes the mapping of the Execute Command procedure call to transport protocol services and the UAS implementation of each transport protocol service.

Table 21 — Execute Command procedure call transport protocol services

Transport protocol service	I/T [a]	Implementation	Reference
Request/Confirmation			
Send SCSI Command request	I	COMMAND IU	7.2
SCSI Command Received indication	T	Receipt of the COMMAND IU	7.3
Send Command Complete response	T	SENSE IU	7.4
Command Complete Received confirmation	I	Receipt of the SENSE IU or problem transmitting the SENSE IU	7.5
Data-in Transfer [b]			
Send Data-in request	T	READ READY IU or ERDY	7.6
Data-in Delivered confirmation	T	Receipt of link layer acknowledgement of the last byte of data transferred	7.7
Data-out Transfer [b]			
Receive Data-out request	T	WRITE READY IU or ERDY	7.8
Data-out Received confirmation	T	Receipt of link layer acknowledgement of the last byte of data transferred	7.9
Terminate Data Transfer [b]			
Terminate Data Transfer request	T		7.10
Data Transfer Terminated confirmation	T		7.11

[a] I/T indicates whether the UAS initiator port (I) or the UAS target port (T) implements the transport protocol service.
[b] Data transfer transport protocol services for SCSI initiator ports are not specified by SAM-4.

An application client requests the processing of a SCSI task management function by invoking SCSI transport protocol services, the collective operation of which is conceptually modeled in the following procedure calls see SAM-4):

a) Service Response = ABORT TASK (IN (Nexus));
b) Service Response = ABORT TASK SET (IN (Nexus));
c) Service Response = CLEAR ACA (IN (Nexus));
d) Service Response = CLEAR TASK SET (IN (Nexus));
e) Service Response = I_T NEXUS RESET (IN (Nexus));
f) Service Response = LOGICAL UNIT RESET (IN (Nexus));
g) Service Response = QUERY TASK (IN (Nexus));
h) Service Response = QUERY TASK SET (IN (Nexus)); and

i) Service Response = QUERY ASYNCHRONOUS EVENT (IN (Nexus), OUT ([Additional Response Information])).

This standard defines the transport protocol services required by SAM-4 in support of these procedure calls. Table 22 describes the mapping of these procedure calls to transport protocol services and the UAS implementation of each transport protocol service.

Table 22 — Execute Command procedure call transport protocol services

Transport protocol service	I/T [a]	Implementation	Reference
Request/Confirmation			
Send Task Management Request request	I	TASK MANAGEMENT IU	7.12
Task Management Request Received indication	T	Receipt of the TASK MANAGEMENT IU	7.13
Task Management Function Executed response	T	RESPONSE IU	7.14
Received Task Management Function Executed confirmation	I	Receipt of the RESPONSE IU or problem transmitting the COMMAND IU	7.15

[a] I/T indicates whether the UAS initiator port (I) or the UAS target port (T) implements the transport protocol service.

7.2 Send SCSI Command transport protocol service

An application client uses the Send SCSI Command transport protocol service request to request that a UAS initiator port transmit a COMMAND IU on the Command pipe.

Send SCSI Command (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-in Buffer Size], [Data-out Buffer], [Data-out Buffer Size], [Command Priority], [CRN]))

Table 23 shows how the arguments to the Send SCSI Command transport protocol service are used.

Table 23 — Send SCSI Command transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to send the COMMAND IU; b) T specifies the target port to which the COMMAND IU is to be sent; c) L specifies the LOGICAL UNIT NUMBER field in the COMMAND IU; and d) Q specifies the TAG field in the COMMAND IU.
CDB	Specifies the CDB field in the COMMAND IU.
Task Attribute	Specifies the TASK ATTRIBUTE field in the COMMAND IU
[Data-in Buffer Size]	Maximum of 2^{32} bytes.
[Data-out Buffer]	Internal to the UAS initiator port.
[Data-out Buffer Size]	Maximum of 2^{32} bytes.
[CRN]	Ignored
[Command Priority]	Specifies the COMMAND PRIORITY field in the COMMAND IU.

7.3 SCSI Command Received transport protocol service

A UAS target port uses the SCSI Command Received transport protocol service indication to notify a task manager that it has received a COMMAND IU.

SCSI Command Received (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Command Priority], [CRN]))

Table 24 shows how the arguments to the SCSI Command Received transport protocol service are determined.

Table 24 — SCSI Command Received transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port from which the COMMAND IU was received; b) T indicates the target port which received the COMMAND IU; c) L indicates the LOGICAL UNIT NUMBER field in the COMMAND IU; and d) Q indicates the TAG field in the COMMAND IU.
CDB	Indicates the CDB field in the COMMAND IU.
Task Attribute	Indicates the TASK ATTRIBUTE field in the COMMAND IU.
[CRN]	Ignored
[Command Priority]	Indicates the COMMAND PRIORITY field in the COMMAND IU.

7.4 Send Command Complete transport protocol service

A device server uses the Send Command Complete transport protocol service response to request that a UAS target port transmit a SENSE IU on the Status pipe.

Send Command Complete (IN (I_T_L_Q Nexus, [Sense Data], [Sense Data Length], Status, [Status Qualifier], Service Response))

A device server shall only call Send Command Complete () after receiving SCSI Command Received ().

A device server shall not call Send Command Complete () for a given I_T_L_Q nexus until:

a) all its outstanding Receive Data-Out () calls for that I_T_L_Q nexus have been responded to with Data-Out Received (); and
b) all its outstanding Send Data-In () calls for that I_T_L_Q nexus have been responded to with Data-In Delivered ().

Table 25 shows how the arguments to the Send Command Complete transport protocol service are used.

Table 25 — Send Command Complete transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to send the SENSE IU; b) T specifies the target port to which the SENSE IU is to be sent; c) L specifies the LOGICAL UNIT NUMBER field in the SENSE IU; and d) Q specifies the TAG field in the SENSE IU.
[Sense Data]	If a SENSE IU is returned, then [Sense Data] specifies the SENSE DATA field in the SENSE IU. If a RESPONSE IU is returned, then [Sense Data] is ignored.
[Sense Data Length]	If a SENSE IU is returned, then [Sense Data Length] specifies LENGTH field in the SENSE IU. If a RESPONSE IU is returned, then [Sense Data Length] is zero.
Status	f a SENSE IU is returned, then Status specifies the STATUS field in the SENSE IU. If a RESPONSE IU is returned, then Status is ignored.
[Status Qualifier]	Ignored
Service Response	If a SENSE IU is returned, then Service Response is COMMAND COMPLETE. If a RESPONSE IU is returned, then Service Response is the RESPONSE CODE field in the RESPONSE IU.

7.5 Command Complete Received transport protocol service

A UAS initiator port uses the Command Complete Received transport protocol service confirmation to notify an application client that it has received a response for its COMMAND IU (e.g., a SENSE IU).

Command Complete Received (IN (I_T_L_Q Nexus, [Data-in Buffer], [Sense Data], [Sense Data Length], Status, [Status Qualifier], Service Response))

Table 26 shows how the arguments to the Command Complete Received transport protocol service are determined.

Table 26 — Command Complete Received transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to send the SENSE IU; b) T specifies the target port to which the SENSE IU is to be sent; c) L specifies the LOGICAL UNIT NUMBER field in the SENSE IU; and d) Q specifies the TAG field in the SENSE IU.
[Data-in Buffer]	Internal to the UAS initiator port.
[Sense Data]	If a SENSE IU is returned, then [Sense Data] indicates the contents of the SENSE DATA field in the SENSE IU. If a RESPONSE IU is returned, then [Sense Data] is ignored.
[Sense Data Length]	If a SENSE IU is returned, then [Sense Data Length] indicates the contents of the LENGTH field in the SENSE IU. If a RESPONSE IU is returned, then [Sense Data Length] is zero.
Status	If a SENSE IU is returned, then Status indicates the the contents of the STATUS field in the SENSE IU. If a RESPONSE IU is returned, then status is ignored.
[Status Qualifier]	Ignored
Service Response	If a SENSE IU is returned, then Service Response is COMMAND COMPLETE. If a RESPONSE IU is returned, then Service Response indicates the contents of the the RESPONSE CODE field in the RESPONSE IU.

7.6 Send Data-In transport protocol service

A device server uses the Send Data-In transport protocol service request to request that a UAS target port transmit read data on the Data-in pipe.

Send Data-In (IN (I_T_L_Q Nexus, Device Server Buffer, Request Byte Count))

A device server shall only call Send Data-In () during a read or bidirectional command.

A device server shall not call Send Data-In () for a given I_T_L_Q nexus after it has called Send Command Complete () for that I_T_L_Q nexus (e.g., a STATUS IU with for that I_T_L_Q nexus) or called Task Management Function Executed for a task management function that terminates that task (e.g., an ABORT TASK).

Table 27 shows how the arguments to the Send Data-In transport protocol service are used.

Table 27 — Send Data-In transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port which the read data is to be sent; b) T specifies the target port to which the read data is to be sent; c) L specifies the logical unit to which the read data is to be sent; and d) Q specifies the tag to which the read data is to be sent.
Device Server Buffer	Internal to the device server.
Request Byte Count	Specifies the length of the read data specified by the command.

7.7 Data-In Delivered transport protocol service

A UAS target port uses the Data-In Delivered transport protocol service indication to notify a device server of the results of transmitting read data.

Data-In Delivered (IN (I_T_L_Q Nexus, Delivery Result))

Table 28 shows how the arguments to the Data-In Delivered transport protocol service are determined.

Table 28 — Data-In Delivered transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port to which the read data has been sent; b) T indicates the target port to which the read data has been sent; c) L indicates the logical unit to which the read data has been sent; and d) Q specifies the tag to which the read data has been sent.
Delivery result	From the response to the outgoing read data: a) DELIVERY SUCCESSFUL: The read data received a USB acknowledgement (see 7.16); or b) DELIVERY FAILURE: All of the data was not successfully received.

7.8 Receive Data-Out transport protocol service

A device server uses the Receive Data-Out transport protocol service request to request that a UAS target port transmit WRITE READY IU on the Status pipe or an ERDY on the Data-out pipe.

Receive Data-Out (IN (I_T_L_Q Nexus, Application Client Buffer Offset, Request Byte Count, Device Server Buffer))

A device server shall only call Receive Data-Out () during a write or bidirectional command.

A device server shall not call Receive Data-Out () for a given I_T_L_Q nexus after a Send Command Complete () has been called for that I_T_L_Q nexus or after a Task Management Function Executed () has been called for a task management function that terminates that command (e.g., an ABORT TASK).

Table 29 shows how the arguments to the Receive Data-Out transport protocol service are used.

Table 29 — Receive Data-Out transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I specifies the initiator port to which the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe is to be sent; b) T specifies the target port to which the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe is to be sent; c) L specifies the logical unit to which the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe is to be sent; and d) Q specifies the tag to which the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe is to be sent.
Application Client Buffer Offset	Ignored
Device Server Buffer	Internal to the device server.
Request Byte Count	Ignored

7.9 Data-Out Received transport protocol service

A UAS target port uses the Data-out Received transport protocol service indication to notify a device server that the attempt to receive write data has completed as the result of transmitting a WRITE READY IU on the Status pipe or ERDY on the Data-out pipe.

Data-out Received (IN (I_T_L_Q Nexus, Delivery Result))

Table 30 shows how the arguments to the Data-out Received transport protocol service are determined.

Table 30 — Data-Out Received transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T_L_Q nexus, where: a) I indicates the initiator port to send the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe; b) T indicates the target port to which the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe is to be sent; c) L indicates the logical unit field in the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe; and d) Q specindicates the tag.
Delivery result	From the response to the WRITE READY IU on the Status pipe or ERDY on the Data-out pipe: a) DELIVERY SUCCESSFUL: The WRITE READY IU on the Status pipe or ERDY on the Data-out pipe was successfully transmitted and all the write data was received; or b) DELIVERY FAILURE: The WRITE READY IU on the Status pipe or ERDY on the Data-out pipe did not receive a USB acknowledgement (see 7.16) or all of the data was not successfully received.

7.10 Terminate Data Transfer transport protocol service

A device server uses the Terminate Data Transfer transport protocol service request to request that a UAS target port terminate any Send Data-In () or Receive Data-Out () transport protocol services, if any, being processed using the specified nexus.

Terminate Data Transfer (IN (Nexus))

Table 31 shows how the arguments to the Terminate Data Transfer transport protocol service are used.

Table 31 — Terminate Data Transfer transport protocol service arguments

Argument	Implementation
nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus, specifying the scope of the data transfer(s) to terminate.

7.11 Data Transfer Terminated transport protocol service

A UAS target port uses the Data Transfer Terminated transport protocol service indication to notify a device server that all data transfers for the indicated nexus have been terminated.

Data Transfer Terminated (IN (Nexus))

Table 32 shows how the arguments to the Data Transfer Terminated transport protocol service are determined.

Table 32 — Data Transfer Terminated transport protocol service arguments

Argument	Implementation
nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus, specifying the scope of the data transfer(s) to terminate.

7.12 Send Task Management Request transport protocol service

An application client uses the Send Task Management Request transport protocol service request to request that a UAS initiator port transmit a TASK MANAGEMENT IU.

Send Task Management Request (IN (Nexus, Function Identifier, Association))

Table 33 shows how the arguments to the Send Task Management Request transport protocol service are used.

Table 33 — Send Task Management Request transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier), where: a) I specifies the initiator port to send the TASK MANAGEMENT IU; b) T specifies the target port to which the TASK MANAGEMENT IU is sent; c) L (for an I_T_L nexus or an I_T_L_Q nexus) specifies the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU; and d) Q (for an I_T_L_Q nexus) specifies the TAG OF TASK TO BE MANAGED field in the TASK MANAGEMENT IU.
Function Identifier	Specifies the TASK MANAGEMENT FUNCTION field in the TASK MANAGEMENT IU. Only these task management functions are supported: a) ABORT TASK (Nexus argument specifies an I_T_L_Q Nexus); b) ABORT TASK SET (Nexus argument specifies an I_T_L Nexus); c) CLEAR ACA (Nexus argument specifies an I_T_L Nexus); d) CLEAR TASK SET (Nexus argument specifies an I_T_L Nexus); e) I_T NEXUS RESET (Nexus argument specifies an I_T Nexus); f) LOGICAL UNIT RESET (Nexus argument specifies an I_T_L Nexus); g) QUERY TASK (Nexus argument specifies an I_T_L_Q Nexus); h) QUERY TASK SET (Nexus argument specifies an I_T_L Nexus); and i) QUERY ASYNCHRONOUS EVENT (Nexus argument specifies an I_T_L Nexus).
Association	Specifies the TAG field in the TASK MANAGEMENT IU.

7.13 Task Management Request Received transport protocol service

A UAS target port uses the Task Management Request Received transport protocol service indication to notify a task manager that it has received a TASK MANAGEMENT IU.

Task Management Request Received (IN (Nexus, Function Identifier, Association))

Table 34 shows how the arguments to the Task Management Request Received transport protocol service are determined.

Table 34 — Task Management Request Received transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier), where: a) I specifies the initiator port to send the TASK MANAGEMENT IU; b) T specifies the target port to which the TASK MANAGEMENT IU is sent; c) L (for an I_T_L nexus or an I_T_L_Q nexus) specifies the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU; and d) Q (for an I_T_L_Q nexus) specifies the TAG OF TASK TO BE MANAGED field in the TASK MANAGEMENT IU.
Function Identifier	Specifies the TASK MANAGEMENT FUNCTION field in the TASK MANAGEMENT IU. Only these task management functions are supported: a) ABORT TASK (Nexus argument specifies an I_T_L_Q Nexus); b) ABORT TASK SET (Nexus argument specifies an I_T_L Nexus); c) CLEAR ACA (Nexus argument specifies an I_T_L Nexus); d) CLEAR TASK SET (Nexus argument specifies an I_T_L Nexus); e) I_T NEXUS RESET (Nexus argument specifies an I_T Nexus); f) LOGICAL UNIT RESET (Nexus argument specifies an I_T_L Nexus); g) QUERY TASK (Nexus argument specifies an I_T_L_Q Nexus); h) QUERY TASK SET (Nexus argument specifies an I_T_L Nexus); and i) QUERY ASYNCHRONOUS EVENT (Nexus argument specifies an I_T_L Nexus).
Association	Specifies the TAG field in the TASK MANAGEMENT IU.

7.14 Task Management Function Executed transport protocol service

A task manager uses the Task Management Function Executed transport protocol service response to request that a UAS target port transmit a RESPONSE IU.

Task Management Function Executed (IN (Nexus, Service Response, [Additional Response Information], Association))

A task manager shall only call Task Management Function Executed () after receiving Task Management Request Received ().

Table 35 shows how the arguments to the Task Management Function Executed transport protocol service are used.

Table 35 — Task Management Function Executed transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier), where: a) I specifies the initiator port to send the TASK MANAGEMENT IU; b) T specifies the target port to which the TASK MANAGEMENT IU is sent; c) L (for an I_T_L nexus or an I_T_L_Q nexus) specifies the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU; and d) Q (for an I_T_L_Q nexus) specifies the TAG OF TASK TO BE MANAGED field in the TASK MANAGEMENT IU.
Service Response	Specifies the DATAPRES field and RESPONSE CODE field in the RESPONSE IU: a) FUNCTION COMPLETE: The RESPONSE IU RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION COMPLETE; b) FUNCTION SUCCEEDED: The RESPONSE IU RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION SUCCEEDED; c) FUNCTION REJECTED: The RESPONSE IU RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED; d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE IU RESPONSE CODE field is set to INCORRECT LOGICAL UNIT NUMBER; or e) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE IU RESPONSE CODE field is set to: A) TASK MANAGEMENT FUNCTION FAILED; or B) OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED..
[Additional Response Information]	Specifies the ADDITIONAL RESPONSE INFORMATION field in the RESPONSE IU.
Association	Specifies the TAG field in the TASK MANAGEMENT IU.

7.15 Received Task Management Function Executed transport protocol service

A UAS initiator port uses the Received Task Management Function Executed transport protocol service confirmation to notify an application client that it has received a response to a TASK MANAGEMENT IU (e.g., received a RESPONSE IU).

Received Task Management Function Executed (IN (Nexus, Service Response, [Additional Response Information], Association))

Table 36 shows how the arguments to the Received Task Management Function Executed transport protocol service are determined.

Table 36 — Received Task Management Function Executed transport protocol service arguments

Argument	Implementation
I_T_L_Q nexus	I_T nexus, I_T_L nexus, or I_T_L_Q nexus (depending on the Function Identifier), where: a) I specifies the initiator port to send the TASK MANAGEMENT IU; b) T specifies the target port to which the TASK MANAGEMENT IU is sent; c) L (for an I_T_L nexus or an I_T_L_Q nexus) specifies the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU; and d) Q (for an I_T_L_Q nexus) specifies the TAG OF TASK TO BE MANAGED field in the TASK MANAGEMENT IU.
Service Response	Specifies the DATAPRES field and RESPONSE CODE field in the RESPONSE IU: a) FUNCTION COMPLETE: The RESPONSE IU RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION COMPLETE; b) FUNCTION SUCCEEDED: The RESPONSE IU RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION SUCCEEDED; c) FUNCTION REJECTED: The RESPONSE IU RESPONSE CODE field is set to TASK MANAGEMENT FUNCTION NOT SUPPORTED; d) INCORRECT LOGICAL UNIT NUMBER: The RESPONSE IU RESPONSE CODE field is set to INCORRECT LOGICAL UNIT NUMBER; or e) SERVICE DELIVERY OR TARGET FAILURE: The RESPONSE IU RESPONSE CODE field is set to: A) TASK MANAGEMENT FUNCTION FAILED; or B) OVERLAPPED INITIATOR PORT TRANSFER TAG ATTEMPTED..
[Additional Response Information]	Specifies the ADDITIONAL RESPONSE INFORMATION field in the RESPONSE IU.
Association	Specifies the TAG field in the TASK MANAGEMENT IU.

7.16 USB Acknowledgement

Table 37 defines USB Acknowledgement.

Table 37 — USB Acknowledgement

USB Protocol	Acknowledgement Definition
USB OUT transactions	The transaction is acknowledged when the device transmits an USB ACK packet for the final data packet received from the USB host (see USB-2 or USB-3) [a].
USB IN transactions	The transaction is acknowledged when the device receives an USB ACK packet for the final data packet transmitted to the USB host (see USB-2 or USB-3) [b].

[a] The device shall be prepared to retransmit any of the data preceding the final ACK transmitted to the USB host as part of normal error handling of USB transmission errors.

[b] A device indicates that status is ready after transmitting the ACK for the last data packet. USB-3 implementations will retry the data transfer for any failed data packet before allowing the device to transmit status on the status endpoint because of USB-3 error handling rules for bulk data transfers and endpoint selection (see USB-3).

NOTE 4 — UAS application clients or USB host implementations may encounter conditions in which the status for a given transaction is received before the transaction for the data transfer is indicated complete at the USB host level. This is due to USB status being transmitted across a different endpoint than the data transfer and the allowance in USB for an endpoint, under some circumstances, to continue to transfer data while error recovery is occurring on a different endpoint in the same device.

8 Device server error handling

If a UAS target port calls Data-Out Received () with a Delivery Result set to a value in table 38, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set as indicated in table 38.

Table 38 — Delivery Result to additional sense code mapping

Delivery Result	Additional sense code
DELIVERY FAILURE - TOO MUCH WRITE DATA	TOO MUCH WRITE DATA
DELIVERY FAILURE - INFORMATION UNIT TOO SHORT	INFORMATION UNIT TOO SHORT
DELIVERY FAILURE - INITIATOR RESPONSE TIMEOUT	INITIATOR RESPONSE TIMEOUT