InterNational Committee for Information Technology Standards (INCITS)
Secretariat: Information Technology Industry Council (ITI)
700 K Street NW, Suite 600, Washington, DC 20001
www.INCITS.org

eb-2020-00882

Document Date:	12/16/2020
To:	INCITS Members
Reply To:	B. Bennett
Subject:	Public Review and Comments Register for the Approval of:
	INCITS 572-202x - Information technology - USB Attached SCSI - 3 (UAS-3)
Due Date:	**The public review is from January 1, 2021 to March 2, 2021.**
Action:	The InterNational Committee for Information Technology Standards (INCITS) announces that the subject-referenced document(s) is being circulated for a 60-day public review and comment period. Comments received during this period will be considered and answered. Commenters who have objections/suggestions to this document should so indicate and include their reasons.

All comments should be forwarded not later than the date noted above to the following address:

INCITS Secretariat/ITI
700 K Street NW - Suite 600
Washington DC 20001
Email: comments@standards.incits.org (preferred)

This public review also serves as a call for patents and any other pertinent issues (copyrights, trademarks). Correspondence regarding intellectual property rights may be emailed to the INCITS Secretariat at patents@itic.org.

dpANS
American National Standard

Project
T10/BSR INCITS 572

Revision 05
November 7, 2020

Information technology -
USB Attached SCSI - 3 (UAS-3)

This is an internal working document of T10, a Technical Committee of Accredited Standards Committee INCITS (International Committee for Information Technology Standards). As such this is not a completed standard and has not been approved. The contents may be modified by the T10 Technical Committee. The contents are actively being modified by T10. This document is made available for review and comment only.

T10 Technical Editor:

Ralph O. Weber
Western Digital
18484 Preston Road
Suite 102, PMB 178
Dallas, TX. 75252
USA

Telephone: 214-912-1373
Email: Ralph.Weber@wdc.com

Reference number
ISO/IEC n/a
INCITS 572-202x

Points of Contact

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

T10 Chair
William Martin
Samsung Semiconductor, Inc
7213 Marblethorpe Drive
Roseville, CA 95747
USA

Telephone: (408) 499-1839
Email: bill.martin@samsung.com

T10 Vice-Chair
Curtis Ballard
Hewlett Packard Enterprise
3404 East Harmony Road
Fort Collins CO 80528
USA

Telephone: (970) 898-3013
Email: curtis.ballard@hpe.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
1101 K Street, NW
Suite 610
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

Purchase INCITS Standards
Web site: http://www.incits.org/standards-information/purchase-standards-or-download-dpans

American National Standard
for Information Technology

USB Attached SCSI - 3

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 - 3 (UAS-3) transport protocol. The UAS-3 transport protocol defines a mechanism to transport SCSI commands using USB hardware. The UAS-3 transport protocol coordinates with other members of the SCSI family of standards via the SAM-6 architecture model. This standard is intended to be used in conjunction with SCSI command set standards and USB specifications.

Published by

American National Standards Institute
25 West 43rd Street 4th floor, New York, New York 10036-7422

Printed in the United States of America

Contents

Page

Tables

Figures

Foreword

This foreword is not part of American National Standard INCITS 572-202x.

The purpose of this standard is to define requirements for the transmission of SCSI commands, in a manner compliant with SAM-6, 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 572-202x, 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, 1101 K Street, NW, Suite 610, Washington, DC 20005.

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:

William Martin, Chair Curtis Ballard, Vice-Chair Curtis Stevens, Secretary

Voting Members

Amphenol Corporation
Mr. Paul Coddington (Principal)
Mr. Brad Brubaker
Mr. Zhineng Fan
Mr. Adrian Green
Mr. Donald Harper
Mr. Yifan Huang
Mr. Sam Kocsis
Mr. Martin Li
Mr. Chris Lyon
Mr. Gregory McSorley
Mr. Raghav Nallan
Mr. Alex Persaud
Mr. Michael Scholeno
Mr. Hu Silver
Mr. Michael Wingard
Mr. CN Wong
Mr. Matt Wright

Voting Members (Continued)

Broadcom Inc.	Mr. Brad Besmer (Principal) Mr. Srikiran Dravida Mr. John Gardner Mr. Jeffrey Gauvin Mr. Rick Kutcipal Mr. Bernhard Laschinsky Mr. David Peterson Mr. James Smart Mr. Jason Stuhlsatz Mr. Steve Wilson
Dell Inc.	Mr. Kevin Marks (Principal) Mr. David Black Mr. George Ericson Mr. Christopher Goonan Mr. Bill Lynn Mr. Marlon Ramroopsingh
Foxconn Electronics	Mr. Fred Fons (Principal) Mr. Gary Hsieh Mr. Glenn Moore Mr. Miller Zhao
Fujitsu America Inc	Mr. Kun Katsumata (Principal) Mr. Osamu Kimura Mr. Mark Malcolm Mr. Gene Owens
Hewlett Packard Enterprise	Mr. Curtis Ballard (Principal) Mr. Chris Cheng Mr. Rob Elliott Mr. Barry Olawsky Mr. Jeff Wolford
IBM Corporation	Mr. Kevin Butt (Principal) Mr. Mike Osborne
Intel Corporation	Mr. Michael Hoard (Principal) Mr. Michale Allison Ms. Kim Malone Mr. Minesh Patel Mr. Eric Pickering Mr. Adee Ran Mr. Tezwen Wang Mr. Juntao Yuan
Kioxia	Mr. John Geldman (Principal) Mr. Cameron Brett Mr. Mark Carlson Mr. Don Harwood Mr. Paul Suhler Mr. Tatsuya Tanaka
Lotes Company Ltd	Mr. D.T. Dao (Principal) Mr. Stephen Chiu Mr. John Lynch
Luxshare-ICT	Mr. Scott Shuey (Principal) Mr. Josue Castillo Mr. Jinhua Chen Mr. Pat Young
Marvell Semiconductor Inc	Mr. Craig Carlson (Principal) Mr. Ke Du Mr. James Walch Mr. Wei Zhou

Voting Members (Continued)

Microsemi - a Microchip company	Mr. Tim Symons (Principal) Mr. Sanjay Goyal Mr. Vincent Hache Mr. David Hong Mr. Adnan Jiwani Mr. Marudhu Karthikeyan Mr. Nitin Kumar Tumkur Suresh Mr. Keith Shaw Mr. Ariel Sibley Mr. Gregory Tabor Mr. Jeremiah Tussey
Microsoft Corporation	Mr. Lee Prewitt (Principal) Mr. Paul Luber Mr. Bryan Matthew Mr. Jose Mannanal Vishal
Molex Inc	Ms. Alex Haser (Principal) Mr. Darian Schulz Mr. Scott Sommers
NetApp Inc	Mr. Frederick Knight (Principal) Mr. Chris Fore Mr. Jaimon George Mr. John Meneghini
Oracle	Mr. Martin Petersen (Principal) Mr. Jon Allen Mr. Hyon Kim Mr. ioannis Psyllas Mr. Lee Wan-Hui
Quantum Corporation	Mr. Darryl Torske (Principal) Mr. Carsten Prigge
Samsung Semiconductor Inc (SSI)	Mr. William Martin (Principal) Ms. Judy Brock Mr. HeeChang Cho Ms. Mikyeong Kang Mr. Sung Lee Mr. Bhavith M.P.
Seagate Technology	Mr. Gerald Houlder (Principal) Mr. Alvin Cox Mr. Ian Davies Mr. John Fleming Mr. Jim Hatfield Mr. Anil Keste Mr. Tony Kilwein Mr. Curtis Stevens Ms. Judy Westby
TE Connectivity	Mr. Matt Schumacher (Principal) Mr. Tom Grzysiewicz Mr. Kyle Klinger Mr. Jeffery Mason Mr. Joel Meyers Mr. Eric Powell Mr. Yasuo Sasaki Mr. Nathan Tracy
Toshiba America Electronic Components Inc	Mr. Patrick Hery (Principal) Dr. Don Harwood Ms. Johanna Hernandez Mr. Scott Wright

Voting Members (Continued)

Viavi Solutions Inc	Mr. Jason Rusch (Principal) Mr. George Bullis Mr. Dominic Coupal Mr. Paul Gentieu Mr. Jason Mann Mr. Francisco Roy Mrs. Yamini Shastry
Western Digital Corporation	Mr. Ralph Weber (Principal) Mr. William Boyle Mr. David Brewer Dr. Jorge Campello Mr. Marvin DeForest Mr. Kirill Dimitrov Mr. Michael Koffman Mr. Dave Landsman Dr. Damien Le Moal Mr. Larry McMillan Mr. Chet Mercado Mr. Nadesan Narenthiran Mr. Christopher Reed Mr. Yoni Shternhell Mr. Eric Richard Zornberg

Advisory Members

Advantest	Ms. Danielle Romano (Principal)
Anritsu Corporation	Mr. Ken Mochizuki (Principal) Mr. Yuji Sekine
BizLink Technology Inc	Mr. Tristan Hsiao (Principal)
Cadence Design Systems, Inc.	Ms. Lana Chan (Principal) Mr. Vinod Lakshman Mr. Deep Mehta Mr. Gurudatta Mewundi
Flextronics International (Taiwan) Ltd	Mr. Roger Wan (Principal) Mr. CC Chou
Futurewei Technologies Inc	Mr. Victor Cao (Principal) Mr. Chaohong (Henry) Hu Mr. Timothy Jeffries
Google	Ms. Radha Ramachandran (Principal) Mr. Thieu Le
JMicron	Mr. Gordon Chang (Principal)
QNAP Systems Inc	Mr. Ming-chih Chang (Principal) Mr. CH Yang
Realtek Semiconductor	Mr. Chien-Kuo Cheng (Principal)
SK Hynix Memory Solutions	Mr. Taeho Lee (Principal) Mr. Jungki Noh Mr. Junho Son
Samtec Inc	Mr. David Givens (Principal)

Advisory Members (Continued)

Silicon Motion Inc	Ms. Amanda Huang (Principal) Mr. Edward Hsieh Mr. Randy Hsu Mr. Sky Hsu
Socionext	Mr. Masaru Suzuki (Principal) Mr. Toru Iwata Mr. Hideyuki Kabuo Mr. Masanori Okinoi
Teledyne LeCroy Corporation	Mr. Amit Bakshi (Principal) Mr. Douglas Lee Mr. Mike Micheletti Mr. Rakesh Kathiresan Natarajanm
UNISYS Corporation	Mr. Jeffrey Dremann (Principal) Mr. Phil Shelton
Ultratest Solutions Ltd	Mr. Jonathan Woodward (Principal) Mr. Mike Mavritsin
VMware Inc	Mr. Neil MacLean (Principal) Mr. Wenchao Cui Mr. George Gao Mr. Wenhau Liu Mr. Mike Panas Mr. Murali Rajagopal Mr. Ahmad Tawil

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., IUs).
Clause 7 describes the SCSI Application Layer Transport Protocol Services.
Bibliography lists a bibliography for this standard.

SCSI standards family

Figure 0 shows the relationship of this standard to the other standards and related projects in the SCSI family of standards as of the publication of this standard.

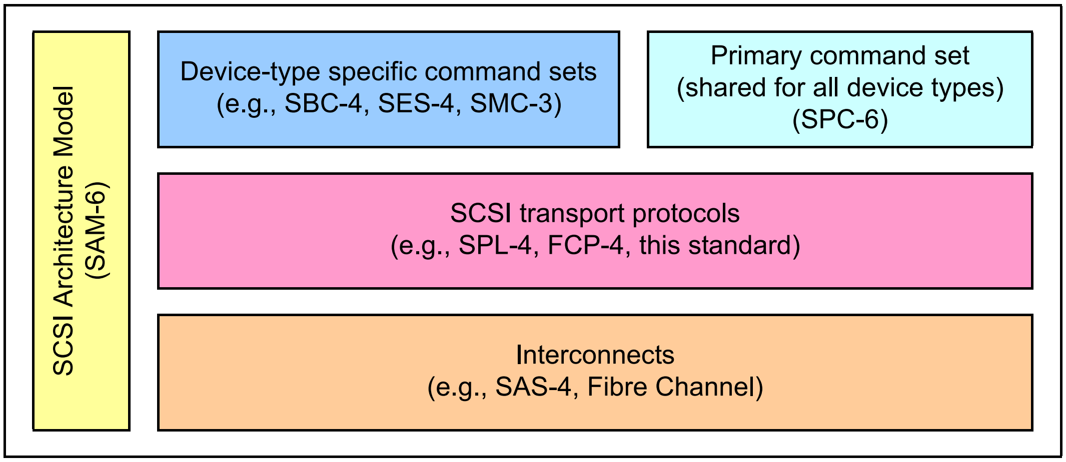


Figure 0 – SCSI document relationships

The SCSI document structure in figure 0 is intended to show the general applicability of the documents to one another. Figure 0 is not intended to imply any hierarchy, protocol stack, or system architecture relationship.

The functional areas identified in figure 0 characterize the scope of standards within a group as follows:

SCSI Architecture Model: Defines the SCSI systems model, the functional partitioning of the SCSI standard set and requirements applicable to all SCSI implementations and implementation standards.

Device-Type Specific Command Sets: Implementation standards that define specific device types including a device model for each device type. These standards specify the required commands and behaviors that are specific to a given device type and prescribe the requirements to be followed by a SCSI initiator device when sending commands to a SCSI target device having the specific device type. The commands and behaviors for a specific device type may include by reference commands and behaviors that are defined by other command sets.

Primary Command Set: An implementation standard that defines a model for all SCSI device types. This standard specifies the required commands and behavior that is common to all SCSI devices, regardless of device type, and prescribes the requirements to be followed by a SCSI initiator device when sending commands to any SCSI target device.

SCSI Transport Protocols: Implementation standards that define the requirements for exchanging information so that different SCSI devices are capable of communicating.

Interconnects: Implementation standards that define the communications mechanism employed by the SCSI transport protocols. These standards may describe the electrical and signaling requirements essential for SCSI devices to interoperate over a given interconnect. Interconnect standards may allow the interconnection of devices other than SCSI devices in ways that are outside the scope of this standard.

The term SCSI is used to refer to the family of standards described in this subclause.

This page intentionally left blank

American National Standard
for Information Technology –

USB Attached SCSI - 3 (UAS-3)

1 Scope

This standard describes a SCSI transport protocol (see SAM-6) for USB-2 and USB-3 with the following properties:

a) a mechanism to send commands associated with any T10 command standard to a USB device;
b) compliance with SCSI Architecture Model - 6 (e.g., autosense and command queuing); and
c) other capabilities.

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this American National Standard. All standards are subject to revision, and parties to agreements based on this American National Standard are encouraged to investigate the possibility of applying the most recent editions of the following standards:

INCITS 546, *Information technology - SCSI Architecture Model - 6 (SAM-6)* (under development)

INCITS 566, *Information technology - SCSI Primary Commands - 6 (SPC-6)* (under development)

Universal Serial Bus Specification Revision 2.0 (USB-2) [1]

Universal Serial Bus 3.2 Specification Revision 1.0 (USB-3) [1]

Universal Serial Bus Mass Storage Class Specification Overview Rev 1.4 (MSC) [1]

1. For information on the current status of USB documents, see the USB Implementors Forum at http://www.usb.org.

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 application client
object that is the source of SCSI commands

Note 1 to entry: See SAM-6.

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

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

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

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

3.1.6 default pipe
USB message pipe created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero

Note 1 to entry: See USB-2.

3.1.7 information unit (IU)
formatted collection of data that carries command information, task management function command information, sense command information, response command information, read ready command information, or write ready command information

Note 1 to entry: See 6.2.

3.1.8 logical unit number (LUN)
64-bit identifier for a logical unit

Note 1 to entry: See SAM-6.

3.1.9 pipe
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

Note 1 to entry: Read data is requested by the Send Data-In transport protocol service (see 7.2.6).

3.1.11 service delivery subsystem
part of the USB I/O system (see USB-3) that transmits information between the UAS initiator port and the UAS target port

Note 1 to entry: USB hubs and USB cables are examples of parts of the USB I/O system that are contained in the service delivery subsystem.

3.1.12 task manager
object that controls the sequencing of commands and processes task management functions

Note 1 to entry: See SAM-6.

3.1.13 transaction packet
header packet used to communicate information between a UAS target device and a UAS initiator device

Note 1 to entry: See USB-3.

3.1.14 UAS domain
one UAS initiator port and one or more UAS target ports

Note 1 to entry: See 4.5.

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

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

3.1.17 UAS target device
USB device that contains one or more UAS target ports

3.1.18 UAS target port
USB interface that contains two USB Bulk-in endpoints (see USB-3), two USB Bulk-out endpoints (see USB-3), and the default USB control endpoint (see USB-3)

3.1.19 USB device
one or more USB interfaces and the default control endpoint

Note 1 to entry: See USB-2 and USB-3.

3.1.20 USB endpoint
collection of characteristics describing the USB device implementation of a pipe

Note 1 to entry: See USB-2 and USB-3.

3.1.21 USB interface
description of one or more USB endpoints

Note 1 to entry: See USB-2 and USB-3.

3.1.22 USB packet
unit of data formatted 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

Note 1 to entry: Write data is requested by the Receive Data-Out transport protocol service (see 7.2.8).

3.2 Symbols and abbreviations

3.2.1 Abbreviations

Abbreviations used in this standard:

Abbreviation	Meaning
IU	Information Unit
LSB	Least significant bit
LUN	Logical unit number
MSB	Most significant bit
MSC	Mass Storage Class (see clause 2)
SAM-6	SCSI Architecture Model - 6
SCSI	Small Computer System Interface
SPC-6	SCSI Primary Commands - 6
UAS	USB Attached SCSI (this standard)
USB	Universal Serial Bus (see USB-2 and USB-3)
USB-2	Universal Serial Bus Revision 2.0 (see clause 2)
USB-3	Universal Serial Bus 3.0 Revision 1.0 (see clause 2)

3.2.2 Units

Units used in this standard:

Units	Meaning
μs	microsecond (i.e., 10^{-6} seconds)
ms	millisecond (i.e., 10^{-3} seconds)
ns	nanosecond (i.e., 10^{-9} seconds)

3.2.3 Mathematical operators

Mathematical operators used in this standard:

Mathematical Operators	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

3.3 Keywords

3.3.1 invalid
keyword used to describe an illegal or unsupported bit, byte, word, field or code value

Note 1 to entry: Receipt by a device server of an invalid bit, byte, word, field or code value shall be reported as error.

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

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

Note 1 to entry: May is synonymous with the phrase “may or may not”.

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

Note 1 to entry: May not is synonymous with the phrase “may or may not”.

3.3.5 obsolete
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

Note 1 to entry: If any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

3.3.7 reserved
keyword referring to bits, bytes, words, fields, and code values that are set aside for future standardization

Note 1 to entry: A reserved bit, byte, word, or field shall be set to zero, or in accordance with a future extension to this standard.

Note 2 to entry: 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
keyword indicating a mandatory requirement

Note 1 to entry: Designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard.

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

Note 1 to entry: Should is synonymous with the phrase “it is strongly recommended”.

3.3.10 vendor specific
something (e.g., a bit, field, code value) that is not defined by this standard

Note 1 to entry: 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 subclause 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 1 shows some examples of decimal numbers using various numbering conventions.

Table 1 – 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 1 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.

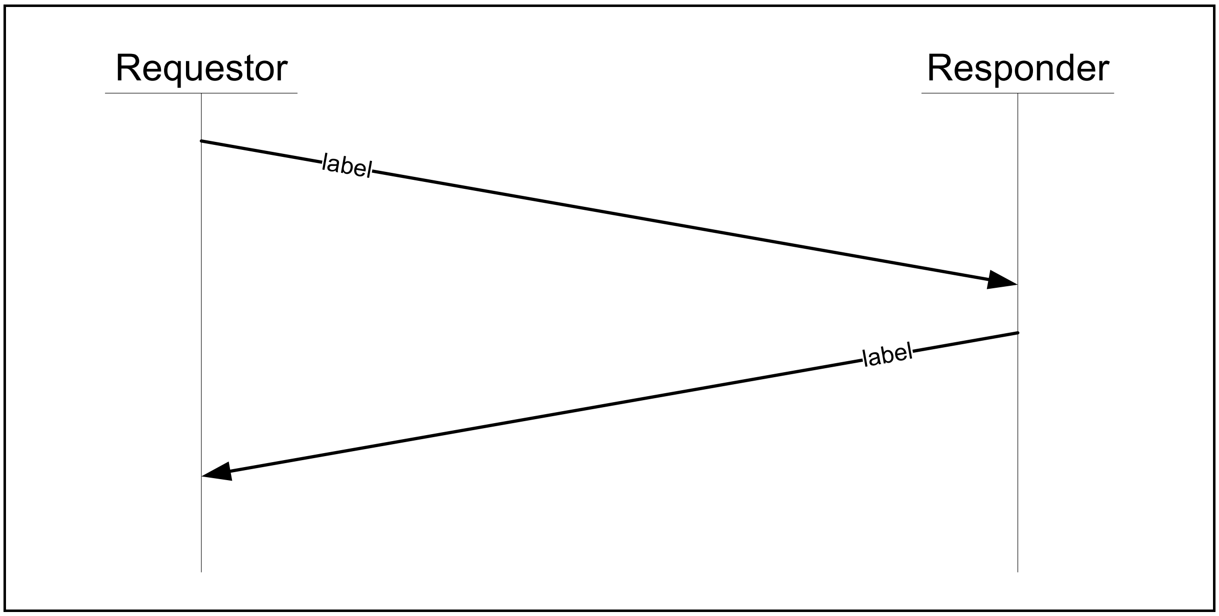


Figure 1 – Example Sequence figure

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 Model overview

A UAS target port shall support a single I_T nexus. The minimum configuration for a UAS target port (see figure 2) 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.

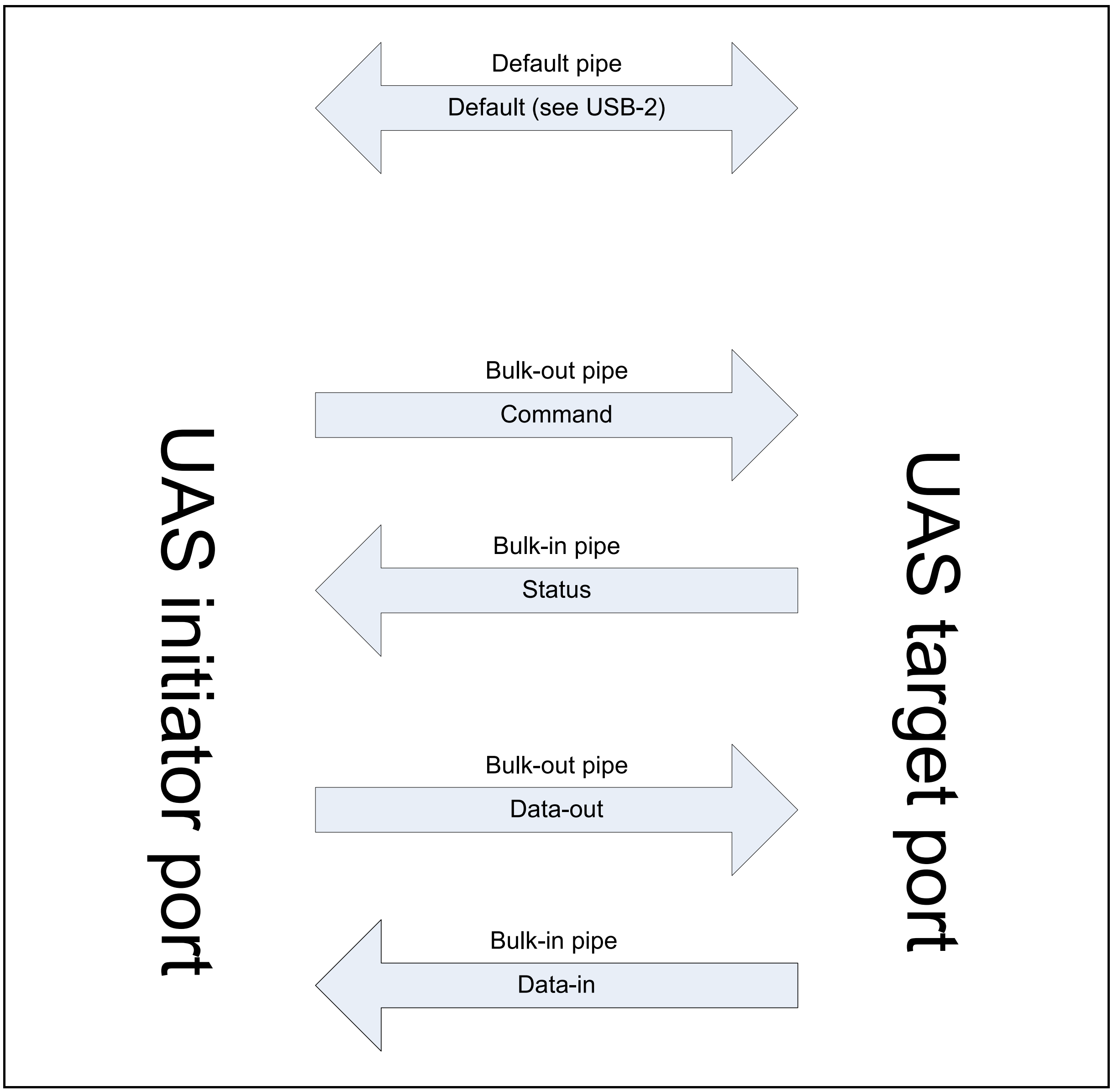


Figure 2 – USB Model

The Default pipe is required by both USB-2 and USB-3, and 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).

Upon the USB device entering the USB configured state (see USB-2 or USB-3):

a) the UAS target port, if any, shall have sufficient buffering or other resources available to receive commands; and
b) the UAS initiator port, if any, should have sufficient buffering available to receive status from the UAS target port.

If the UAS target port is unable to send status to the UAS initiator port when that 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 and command identifiers

4.2.1 Tag handling and command identifiers overview

Every IU defined by the standard (see 6.2) contains a TAG field (i.e., a command identifier (see SAM-6)) in which the value is a unique identifier that the UAS initiator device establishes or has established for:

a) a command (e.g., as specified by a COMMAND IU (see 6.2.2) or a SENSE IU (see 6.2.5)) that the UAS target device is requested to process or is processing; or
b) a task management function (e.g., as specified by a TASK MANAGEMENT IU (see 6.2.7) or a RESPONSE IU (see 6.2.6)) that the UAS target device is requested to process or is processing.

The values that the UAS initiator device specifies in TAG fields are required to be unique across:

a) all logical units in the UAS target device; and
b) all:
   A) commands; and
   B) task management functions,

   for which the UAS initiator device requests concurrent processing.

The UAS target device may perform tag checking as described in 4.2.3 to validate the unique tag values sent by the UAS initiator device.

The TAG field in a COMMAND IU contains the command identifier as defined in SAM-6. The TAG field in a TASK MANAGEMENT IU (see 6.2.7) is an association between a SAM-6 Received Task Management Function Executed transport protocol service (see 7.2.15) and the preceding SAM-6 Send Task Management Request (see 7.2.12).

4.2.2 LUN determination without a LOGICAL UNIT NUMBER field

Not all IUs contain a LOGICAL UNIT NUMBER field.

The COMMAND IU (see 6.2.2) and TASK MANAGEMENT IU (see 6.2.7) contain a LOGICAL UNIT NUMBER field and a TAG field.

In those cases where a LOGICAL UNIT NUMBER field is not present, the contents of the TAG field are used to determine the logical unit number:

a) by the UAS target device for the ERDY transaction packet (see 4.4), the READ_READY_IU (see 6.2.3), and the WRITE_READY_IU (see 6.2.4); and
b) by the UAS initiator device for the SENSE_IU (see 6.2.5) and the RESPONSE_IU (see 6.2.6).

4.2.3 Tag checking

If a UAS target device performs tag checking and a UAS target port calls SCSI Command Received () (see 7.2.3) 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-6.

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 () (see 7.2.13) 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 () (see 7.2.14) with:
    A) the Nexus argument set to I_T nexus;
    B) the Management Identifier argument set to zero; and
    C) the Service Response argument set to SERVICE DELIVERY OR TARGET FAILURE - Overlapped Tag (see table 39).

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 in a contemporaneous manner:

a) transfer data;
b) accept COMMAND IUs (see 6.2.2) and TASK MANAGEMENT IUs (see 6.2.7) 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 (see 6.2.5) 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 (see 6.2.3) or a WRITE READY IU (see 6.2.4) on the Status pipe, then that UAS target device shall be ready to send or receive all the data for the indicated I_T_L nexus and command identifier. Until processing of the Data-In Delivery transport protocol service (see 7.2.7) or the Data-Out Received transport protocol service (see 7.2.9) is completed, the UAS target device shall not use the associated Data-out pipe or Data-in pipe to transfer data for any other I_T_L nexus and command identifier.

After the last byte of data has been transferred and achieved a USB acknowledgment (see 7.2.16), the UAS target device shall return a SENSE IU on the Status pipe to indicate command completion.

The UAS target device may use the associated Data-out pipe to transfer data for a different command upon:

1) receipt of the USB acknowledgement for the transfer of the last byte of data for an I_T_L nexus and command identifier; and
2) sending a WRITE READY IU for the I_T_L nexus and command identifier associated with the data to be transferred on that Data-out pipe.

The UAS target device may use the associated Data-in pipe to transfer data for a different command upon:

1) receipt of the USB acknowledgement for the transfer of the last byte of data for an I_T_L nexus and command identifier; and
2) sending a READ READY IU for the I_T_L nexus and command identifier associated with the data to be transferred on that Data-in pipe.

4.4 USB-3 ERDY transaction packet considerations

As a part of USB-3 SuperSpeed operation, the ERDY transaction packet used by UAS target ports to notify UAS initiator ports that the UAS target device is ready to transfer IUs or user data for the indicated tag (see 4.2). The tag for the command or task management function that is being processed shall be placed in the stream id of the ERDY being sent.

UAS target ports that connect via USB-3 SuperSpeed shall send an ERDY transaction packet on:

a) the Status pipe to indicate that a SENSE IU (see 6.2.5) or RESPONSE IU (see 6.2.6) is about to be transferred;
b) the Data-in pipe to indicate that read data is about to be transferred (see 4.3); and
c) the Data-out pipe to request that the UAS initiator device transfer write data (see 4.3).

If an ERDY transaction packet is sent on:

a) a Data-in pipe, then a READ READY IU (see 6.2.3) shall not be sent for the indicated tag; and
b) a Data-out pipe, then a WRITE READY IU (see 6.2.4) shall not be sent for the indicated tag.

4.5 UAS domain

Figure 3 shows an example of a simple UAS domain that contains:

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

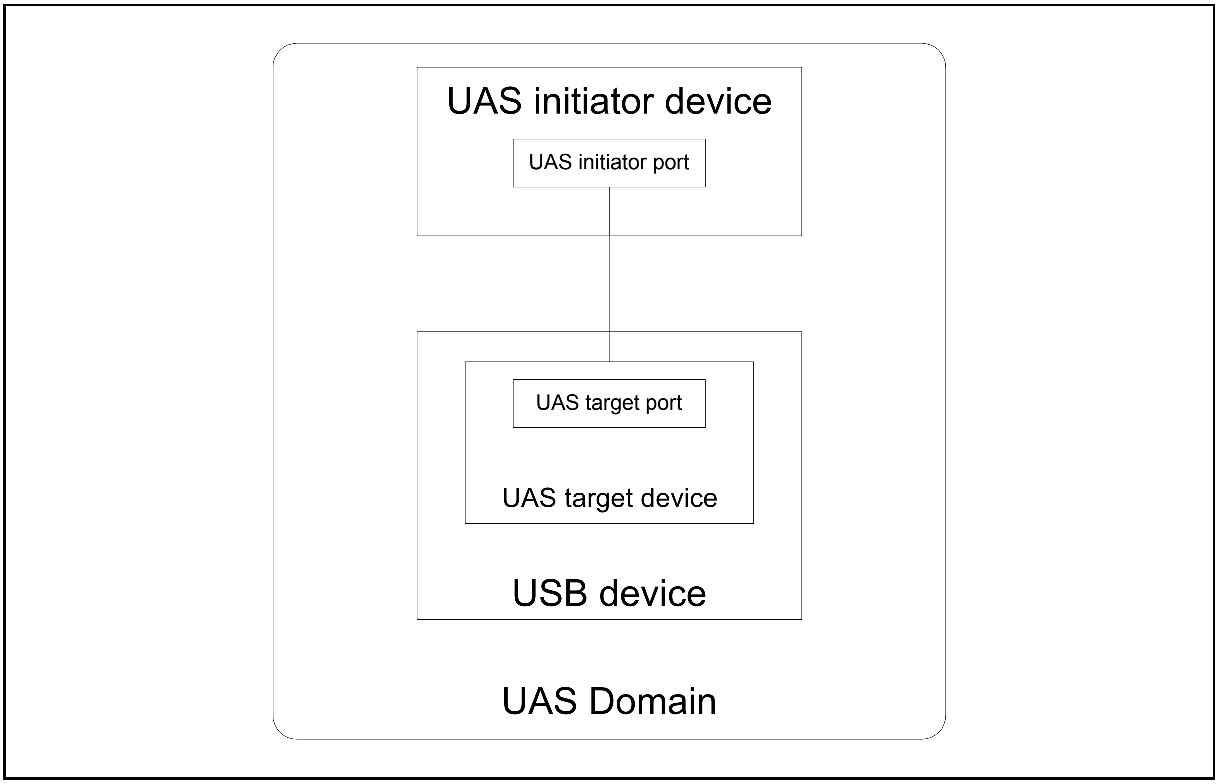


Figure 3 – Example Simple UAS domain

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

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

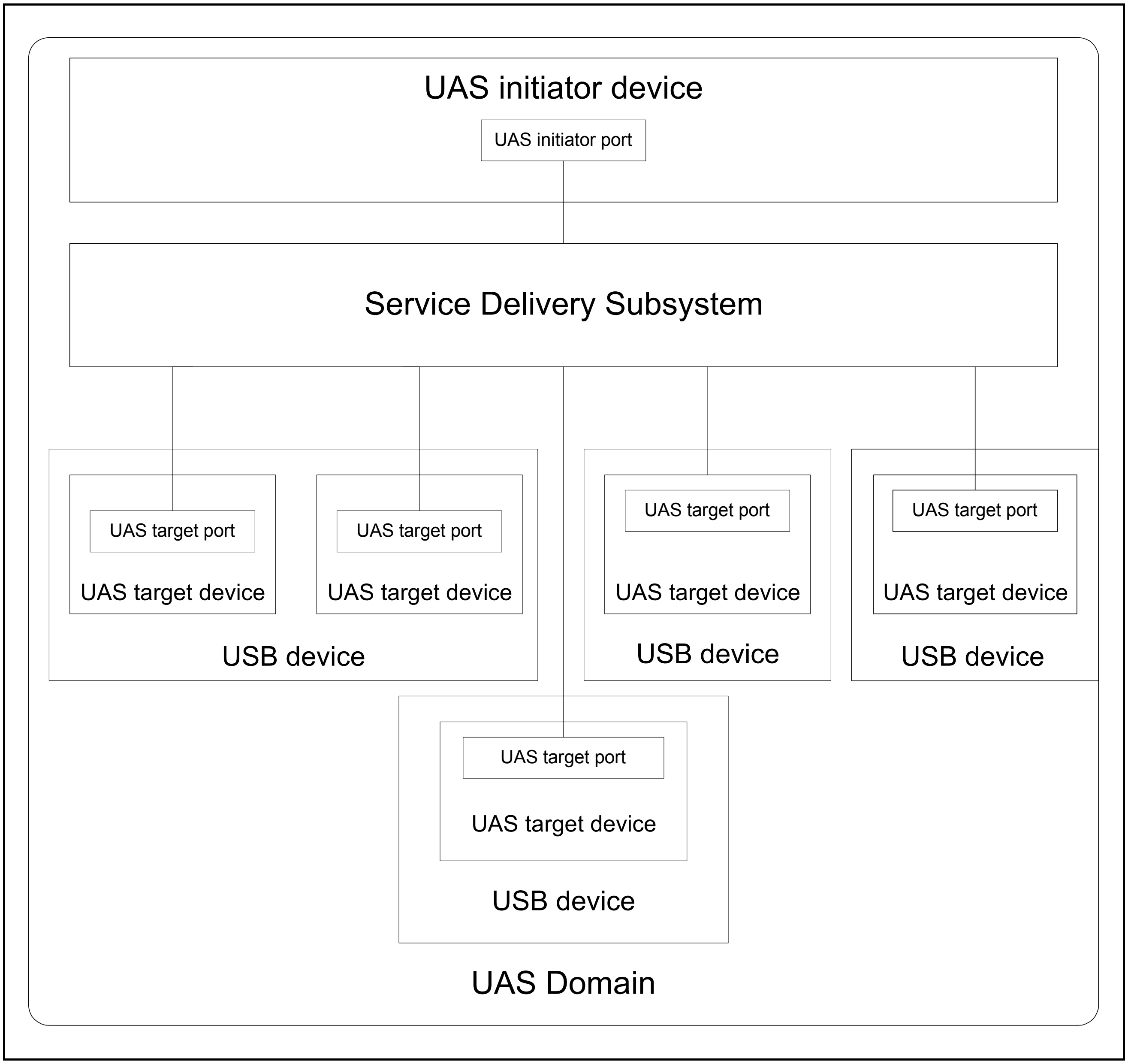


Figure 4 – Example Complex UAS Domain

4.6 Names and identifiers

Table 2 describes the definitions of SCSI architecture object name attributes and identifier attributes for this standard.

Table 2 – SCSI architecture model object name attributes and identifier attributes

Attribute	Definition	Usage in this standard	Reference
SCSI device name	Not specified [a]		
Initiator port name	None [b]		
Target port name	None [b]		
Initiator port identifier	None [c]		
Target port identifier	Device address and interface number	Reported in the Device Identification VPD page (see SPC-6)	USB-2 and USB-3

[a] This standard does not specify a format for this attribute.
[b] SCSI ports compliant with this standard do not implement this attribute.
[c] There is exactly one SCSI initiator port in a UAS domain.

4.7 Resets

A UAS target device shall perform the operations for all reset conditions resulting from SCSI events as defined in SAM-6 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-6).

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-6. 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 USB 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

Byte \ Bit	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

Byte \ Bit	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 format.

Table 6 – Bulk-in endpoint descriptor

Byte \ Bit	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

Byte \ Bit	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

Byte \ Bit	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 to DFh	Reserved
E0h to EFh	Vendor specific
F0h to FFh	Reserved

6 Transport

6.1 Transport overview

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

6.2 IUs

6.2.1 IUs Overview

Table 10 is a summary of the Information Units (IUs) and the associated 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 to FFh	Reserved	

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

Table 11 – IU Header

Byte \ Bit	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 containing a reserved value, then the UAS target port shall call Task Management Function Executed () (see 7.2.14) with:

a) the Nexus argument set to I_T nexus;
b) the Management Identifier argument set to the value in the TAG field; and
c) the Service Response argument set to SERVICE DELIVERY OR TARGET FAILURE - Invalid IU (see table 39).

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

Byte \ Bit	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	(MSB)	ADDITIONAL CDB BYTES						
• • •								
31 + n x 4								(LSB)

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-6.

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-6).
001b	HEAD OF QUEUE	Specifies that the command be managed according to the rules for a head of queue task attribute (see SAM-6).
010b	ORDERED	Specifies that the command be managed according to the rules for an ordered task attribute (see SAM-6).
011b	Reserved	
100b	ACA	Specifies that the command be managed according to the rules for an automatic contingent allegiance task attribute (see SAM-6).
101b to 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-6. If the addressed logical unit does not exist, the task manager shall call Task Management Function Executed () (see 7.2.14) with:

a) the Nexus argument set to I_T nexus;
b) the Management Identifier argument set to the value in the TAG field; and
c) the Service Response argument set to INCORRECT LOGICAL UNIT NUMBER (see table 39).

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-6).

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 transmit an ERDY transaction packet (see 4.4) 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 transmit an ERDY transaction packet (see 4.4) 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	Reserved							
•••								
13								
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-6);

The STATUS field shall be set to the status code (see SAM-6) for the command that has completed.

The 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-6).

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	(MSB)	ADDITIONAL RESPONSE INFORMATION						
•••								
6								(LSB)
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-6. 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 [a]	Command [b]
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 to 07h	Reserved	Invalid	Invalid
08h	TASK MANAGEMENT FUNCTION SUCCEEDED	Valid	Invalid
09h	INCORRECT LOGICAL UNIT NUMBER	Valid	Invalid
0Ah	OVERLAPPED TAG ATTEMPTED [c]	Valid	Valid
0Bh to FFh	Reserved	Invalid	Invalid

[a] The Task column indicates the valid and invalid response codes returned by the UAS target device in response to a TASK MANAGEMENT IU.
[b] The Command column indicates the valid and invalid response codes returned by the UAS target device in response to a COMMAND IU.
[c] Returned in case of command/task management function or task management function/task management function tag conflicts.

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

Byte \ Bit	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. Support for task management functions not listed in table 20 is outside the scope of this standard.

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 LUN [a]	Uses the TOTTBM [b]	Description
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-6). [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-6). [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 determine the task set to be cleared (see SAM-6). [c]
05h to 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-6). [c]
09h to 0Fh	Reserved			
10h	I_T NEXUS RESET	no	no	The task manager shall perform the I_T NEXUS RESET task management function (see SAM-6). [c]
11h to 3Fh	Reserved			
40h	CLEAR ACA	yes	no	The task manager shall perform the CLEAR ACA task management function using the value of the LOGICAL UNIT NUMBER field (see SAM-6). [c]
41h to 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 determine the task to be queried (see SAM-6). [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-6). [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-6). [c]
83h to 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 call Task Management Function Executed () (see 7.2.14) with:

a) the Nexus argument set to I_T nexus;
b) the Management Identifier argument set to the value in the TAG field; and
c) the Service Response argument set to FUNCTION REJECTED (see table 39).

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-6. If the addressed logical unit does not exist, the task manager shall call Task Management Function Executed () (see 7.2.14) with:

a) the Nexus argument set to I_T nexus;
b) the Management Identifier argument set to the value in the TAG field; and
c) the Service Response argument set to INCORRECT LOGICAL UNIT NUMBER (see table 39).

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 5 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.

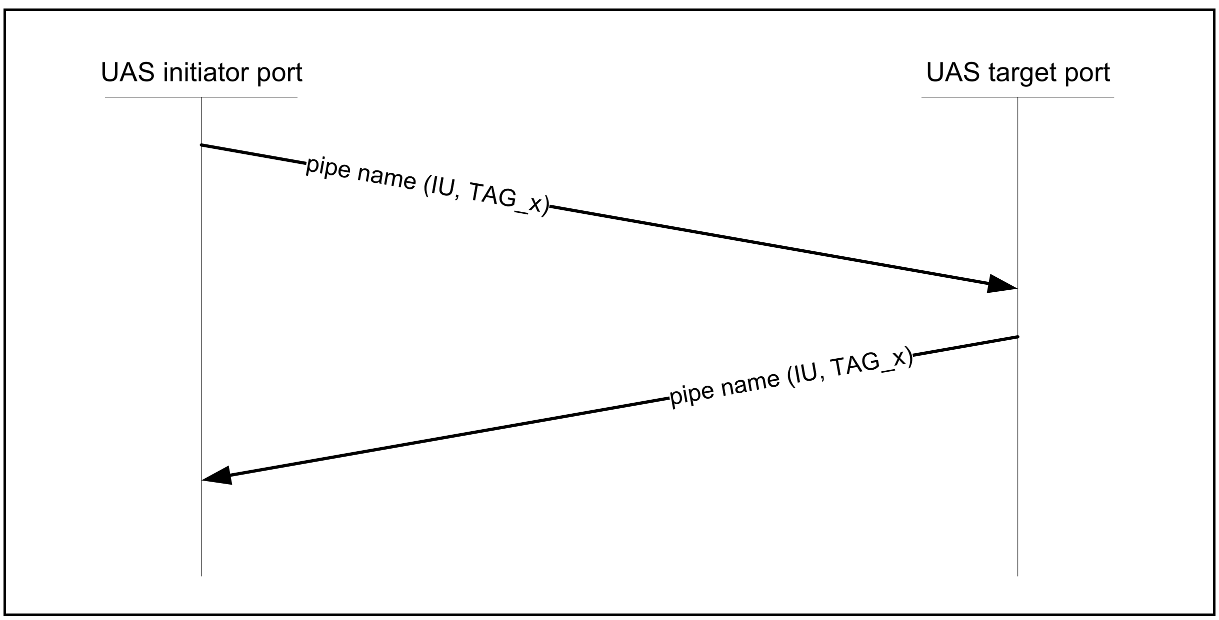


Figure 5 – UAS sequence figure notation

6.3.2 Non-data command/sense sequence

Figure 6 describes the sequence of communication between the UAS initiator port and UAS target port for a command that does not require data transfer.

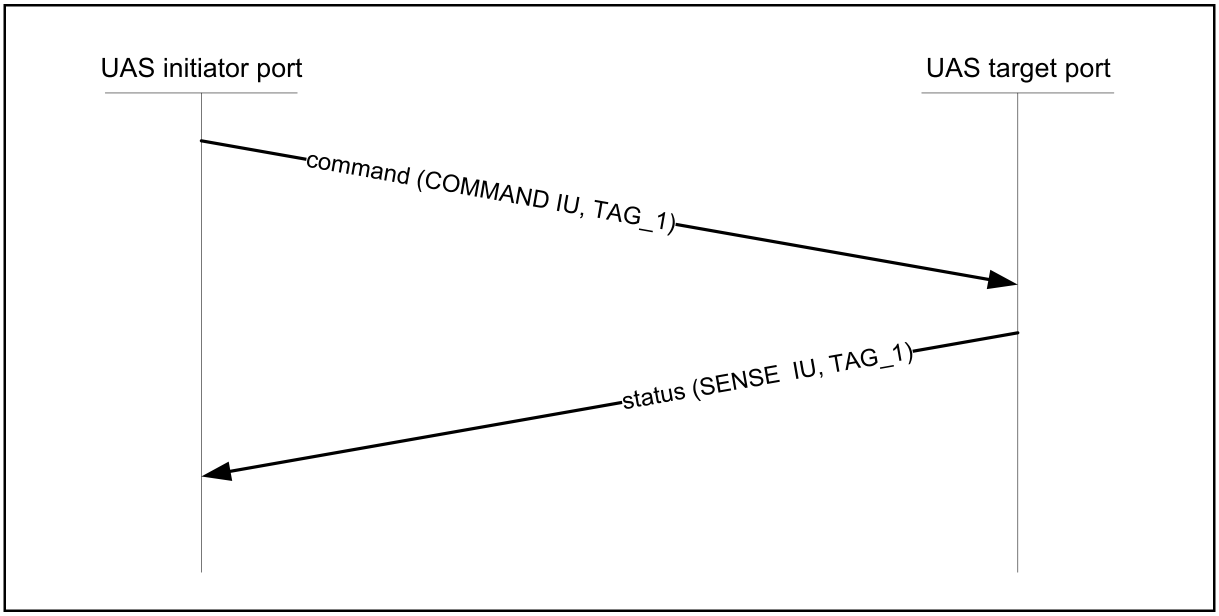


Figure 6 – Non-data transfer with Sense

6.3.3 Non-data command/response sequence

Figure 7 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 a Service Response argument of SERVICE DELIVERY OR TARGET FAILURE - Overlapped Tag (see table 39)).

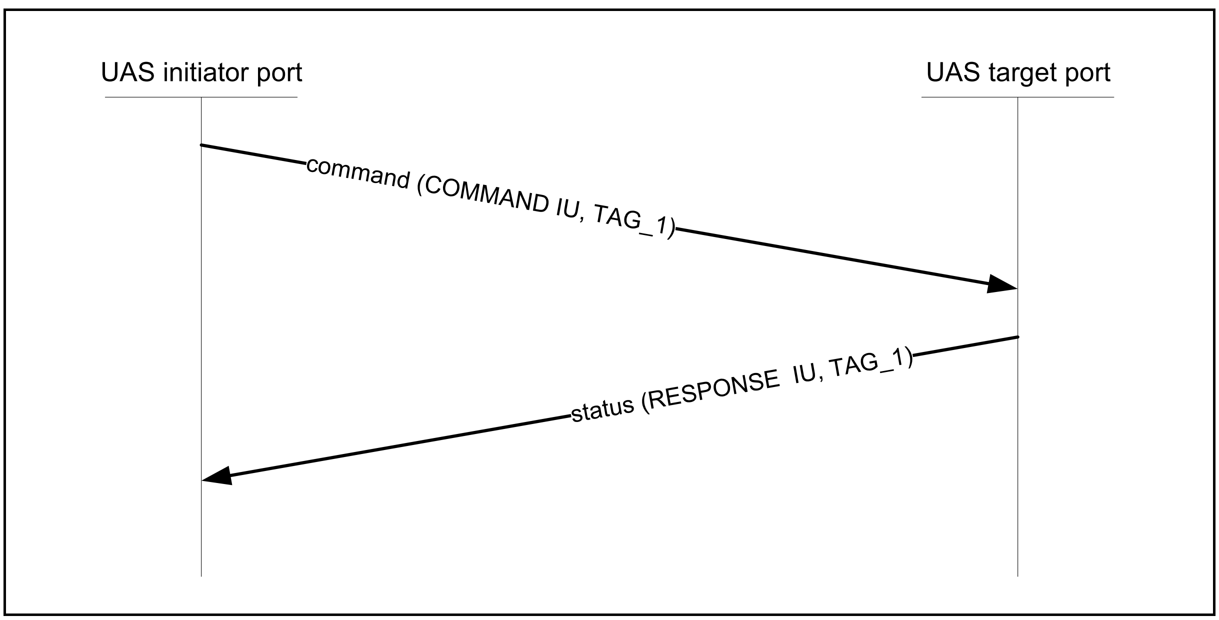
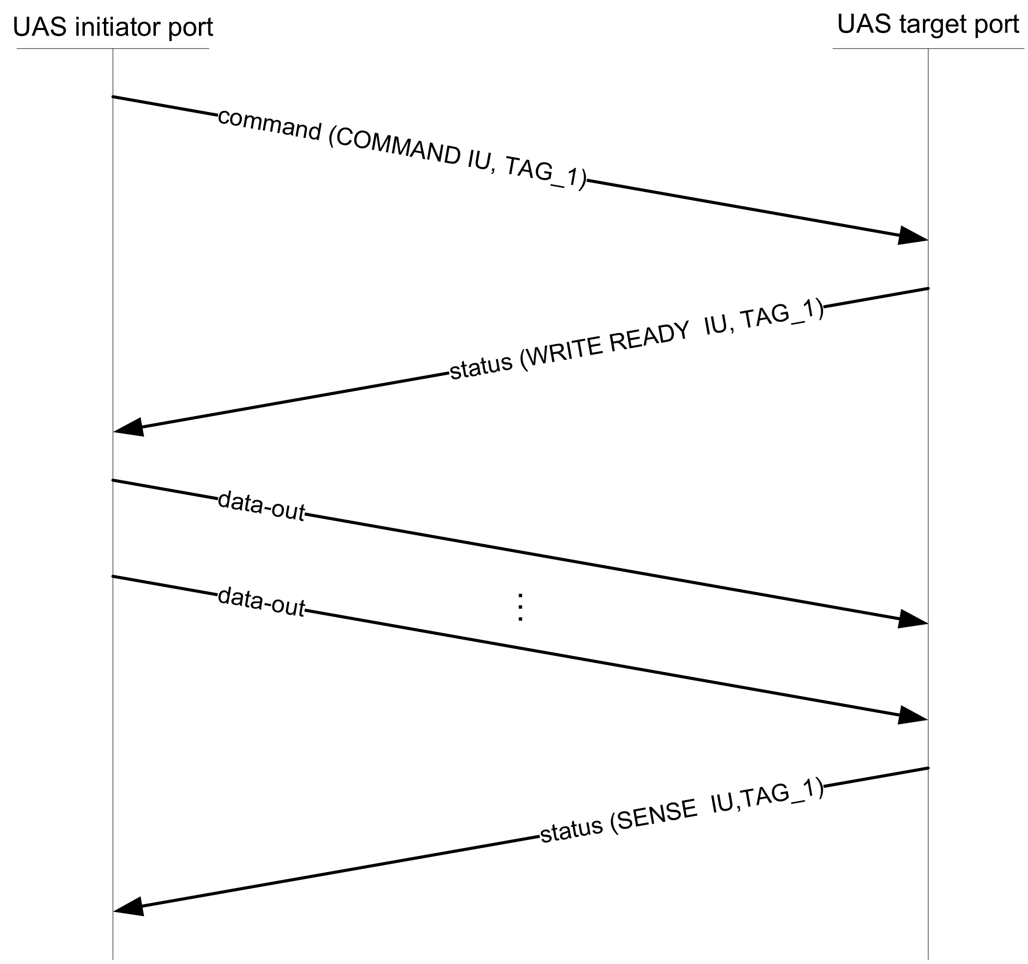


Figure 7 – Non-data Transfer with Response

6.3.4 Data-out command sequence

Figure 8 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).

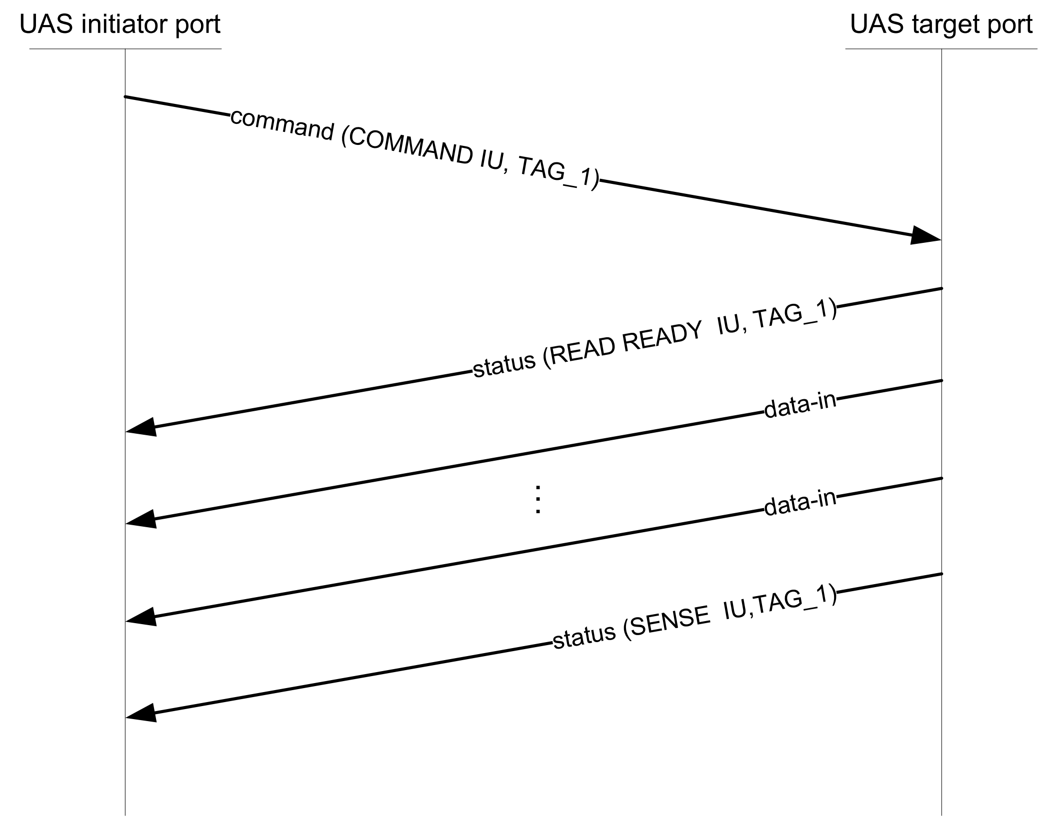


NOTE - The WRITE READY IU is replaced by an ERDY transaction packet (see 4.4) for UAS target ports that connect via SuperSpeed.

Figure 8 – Write Data Transfer

6.3.5 Data-in command sequence

Figure 9 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 READ READY IU is replaced by an ERDY transaction packet (see 4.4) for UAS target ports that connect via SuperSpeed.

Figure 9 – Read Data Transfer

6.3.6 Task management function sequence

Figure 10 describes the sequence of communication between the UAS initiator port and UAS target port for a task management function.

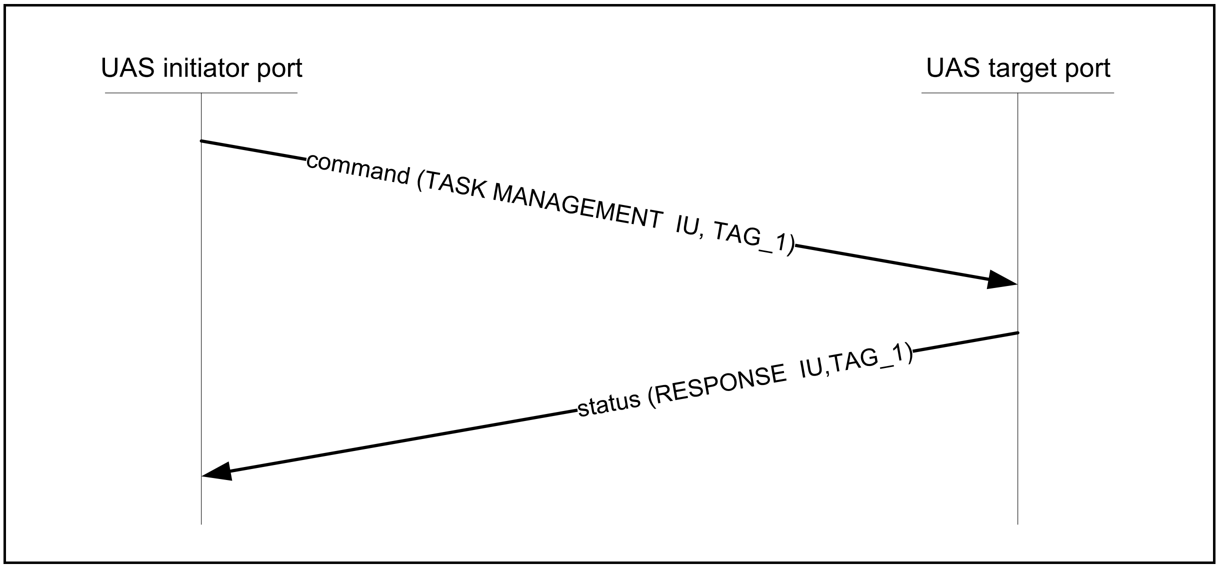
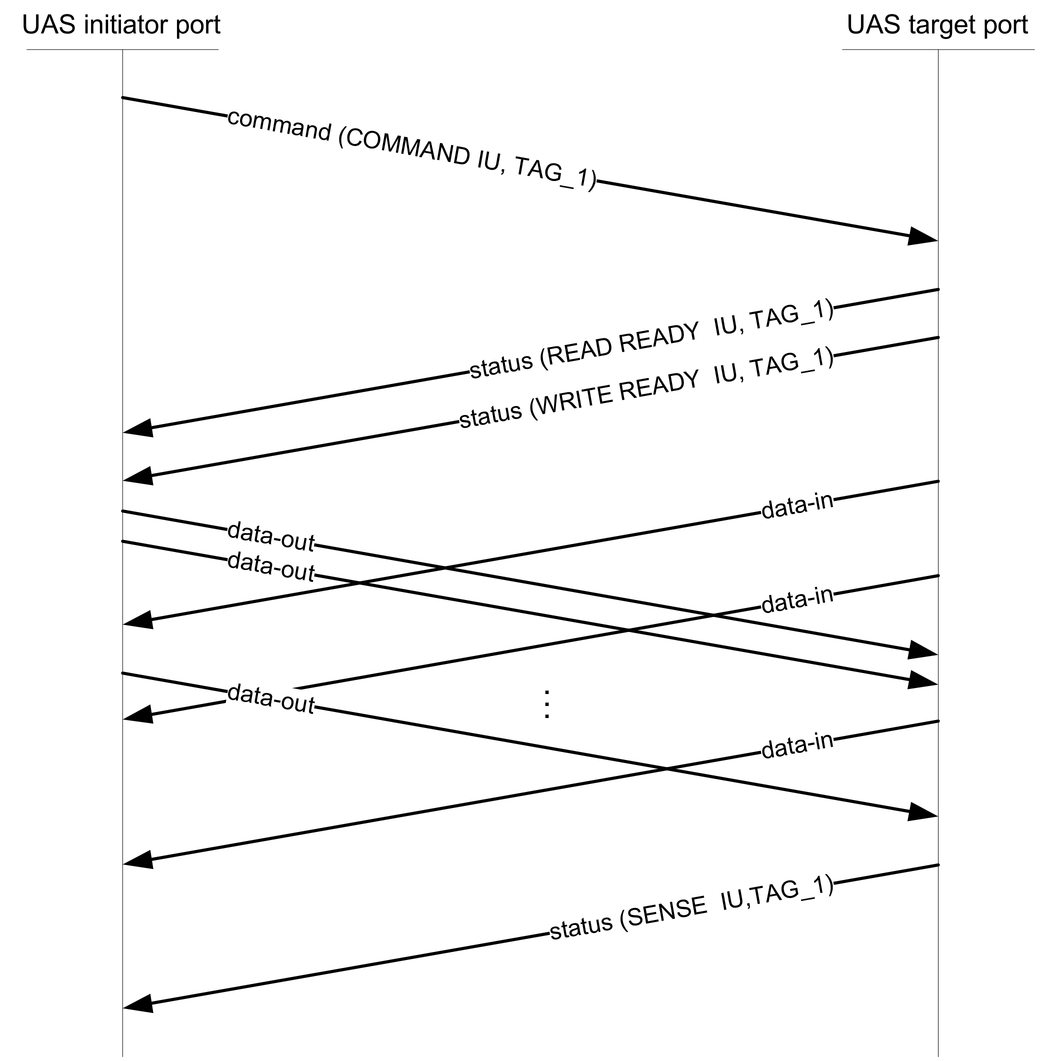


Figure 10 – Task Management

6.3.7 Bi-directional command sequence

Figure 11 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 4.4) for UAS target ports that connect via SuperSpeed.

Figure 11 – Bi-directional Data Transfer

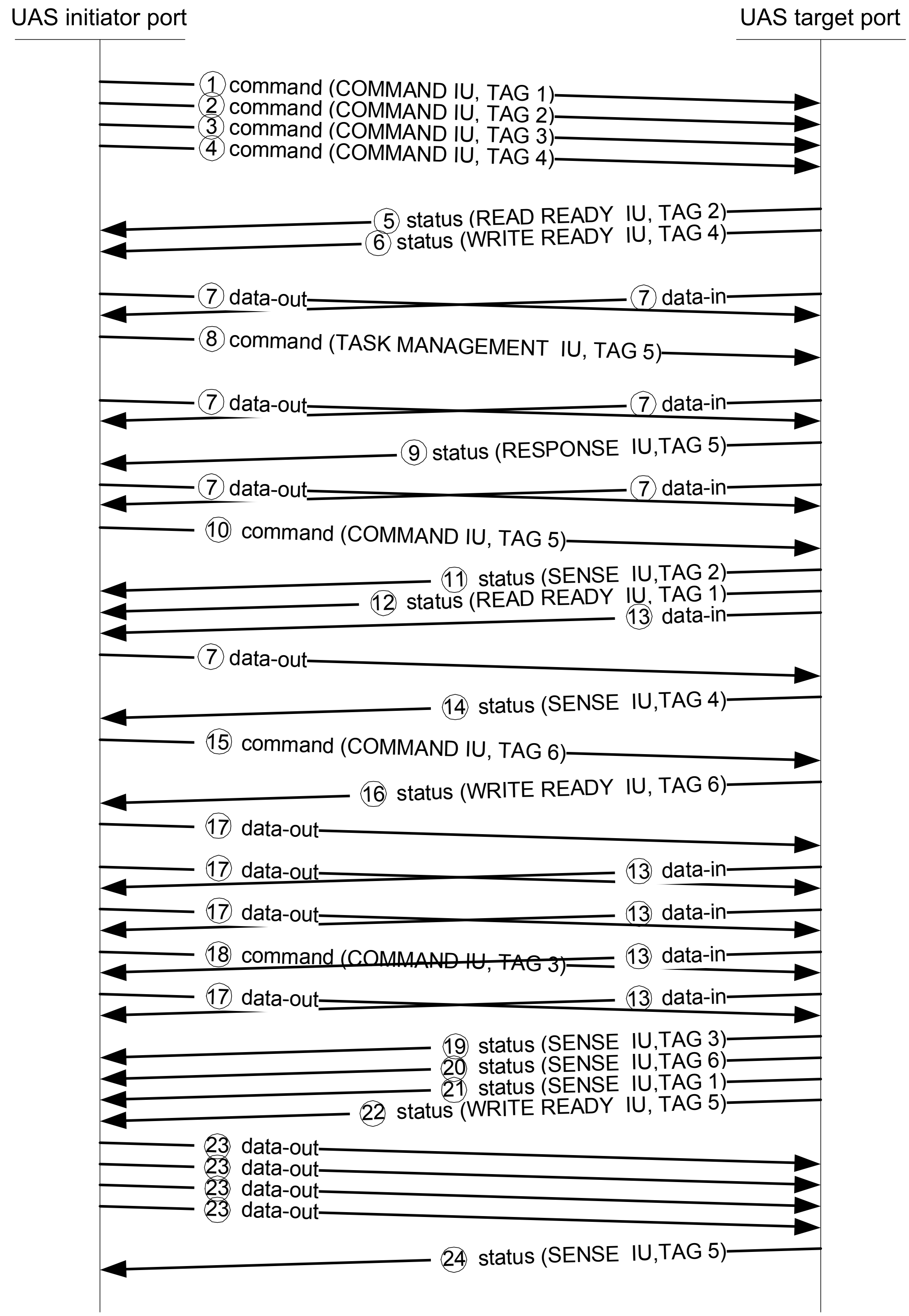
The order of the data-in and data-out phases of a bidirectional command may be influenced by both the definition of the bidirectional command and the capabilities of the USB target.

6.3.8 Multiple command example

Figure 12 describes the sequence of communication between a UAS initiator port and a UAS target port for several commands using task set management (see SAM-6) 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.

In figure 12, step numbers in circles have been inserted in front of the pipe names to match the step numbers in this subclause (i.e., 6.3.8).



NOTE - The WRITE READY IU and the READ READY IU are replaced by an ERDY transaction packet (see 4.4) for UAS target ports that connect via SuperSpeed.

Figure 12 – Multiple Command Example

7 SCSI Application Layer

7.1 Device Identification VPD page

In the Device Identification VPD page returned in response to an INQUIRY command (see SPC-6), each logical unit in a UAS target device shall include the designation descriptors for the target port identifier (see 4.6) and the relative target port identifier (see SAM-6 and SPC-6) listed in table 21.

Table 21 – Device Identification VPD page designation descriptors for the UAS target port

Field in designation descriptor	Designation descriptor	
	Target port identifier	**Relative target port identifier**
DESIGNATOR TYPE	9h (i.e., USB target port identifier)	4h (i.e., relative target port identifier)
ASSOCIATION	01b (i.e., SCSI target port)	01b (i.e., SCSI target port)
CODE SET	1h (i.e., binary)	1h (i.e., binary)
DESIGNATOR LENGTH	4	4
PIV (protocol identifier valid)	1	1
PROTOCOL IDENTIFIER	9h (i.e., UAS)	9h (i.e., UAS)
DESIGNATOR	USB device address and USB interface number [a] (see 4.6)	Relative port identifier [b] as described in SAM-6 and SPC-6

[a] The DESIGNATOR field contains the USB device address of the UAS target port through which the INQUIRY command was received.
[b] The DESIGNATOR field contains the relative port identifier of the UAS target port through which the INQUIRY command was received.

The Device Identification VPD page (see SPC-6) 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.

7.2 Transport Protocol Services

7.2.1 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-6):

Service response = Execute Command (IN (I_T_L Nexus, Command Identifier, 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-6 in support of these procedure calls. Table 22 describes the mapping of the Execute Command procedure call 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 SCSI Command request	I	COMMAND IU	7.2.2
SCSI Command Received indication	T	Receipt of the COMMAND IU	7.2.3
Send Command Complete response	T	SENSE IU	7.2.4
Command Complete Received confirmation	I	Receipt of the SENSE IU or problem transmitting the SENSE IU	7.2.5
Data-in Transfer [b]			
Send Data-in request	T	READ READY IU or ERDY	7.2.6
Data-in Delivered confirmation	T	Receipt of link layer acknowledgement of the last byte of data transferred	7.2.7
Data-out Transfer [b]			
Receive Data-out request	T	WRITE READY IU or ERDY	7.2.8
Data-out Received confirmation	T	Receipt of link layer acknowledgement of the last byte of data transferred	7.2.9
Terminate Data Transfer [b]			
Terminate Data Transfer request	T		7.2.10
Data Transfer Terminated confirmation	T		7.2.11

[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-6.

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-6):

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 task management function transport protocol services required by SAM-6 in support of the task management function procedure calls. Table 23 describes the mapping of these procedure calls to transport protocol services and the UAS implementation of each transport protocol service.

Table 23 – Task management function procedure call transport protocol services

Transport protocol service	I/T [a]	Implementation	Reference
Request/Confirmation			
Send Task Management Request request	I	TASK MANAGEMENT IU	7.2.12
Task Management Request Received indication	T	Receipt of the TASK MANAGEMENT IU	7.2.13
Task Management Function Executed response	T	RESPONSE IU	7.2.14
Received Task Management Function Executed confirmation	I	Receipt of the RESPONSE IU or problem transmitting the COMMAND IU	7.2.15 and 7.2.5

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

7.2.2 Send SCSI Command transport protocol service

An application client uses the Send SCSI Command transport protocol service to request that a UAS initiator port send a COMMAND IU (see 6.2.2) on the Command pipe.

Send SCSI Command (IN (I_T_L Nexus, Command Identifier, CDB, Task Attribute, [Data-in Buffer Size], [Data-out Buffer], [Data-out Buffer Size], [CRN], [Command Priority], [First Burst Enabled]))

Table 24 shows how the arguments to the Send SCSI Command transport protocol service are used.

Table 24 – Send SCSI Command transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: a) I specifies the initiator port used to send the COMMAND IU; b) T specifies the target port to which the COMMAND IU is to be sent; and c) L specifies the value used to set the LOGICAL UNIT NUMBER field in the COMMAND IU.
Command Identifier	Specifies the value used to set the TAG field in the COMMAND IU.
CDB	Specifies the CDB field in the COMMAND IU.
Task Attribute	Specifies the value used to set 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 value used to set the COMMAND PRIORITY field in the COMMAND IU.
[First Burst Enabled]	Ignored

7.2.3 SCSI Command Received transport protocol service

A UAS target port uses the SCSI Command Received transport protocol service to notify a task manager that a COMMAND IU (see 6.2.2) has been received.

SCSI Command Received (IN (I_T_L Nexus, Command Identifier, CDB, Task Attribute, [CRN], [Command Priority]), [First Burst Enabled])

Table 25 shows how the arguments to the SCSI Command Received transport protocol service are determined.

Table 25 – SCSI Command Received transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: a) I specifies the initiator port from which the COMMAND IU was received; b) T specifies the target port that received the COMMAND IU; and c) L specifies the contents of the LOGICAL UNIT NUMBER field in the COMMAND IU.
Command Identifier	Specifies the contents of the TAG field in the COMMAND IU.
CDB	Specifies the contents of the CDB field in the COMMAND IU.
Task Attribute	Specifies the contents of the TASK ATTRIBUTE field in the COMMAND IU.
[CRN]	Ignored
[Command Priority]	Specifies the contents of the COMMAND PRIORITY field in the COMMAND IU.
[First Burst Enabled]	Ignored

7.2.4 Send Command Complete transport protocol service

A device server uses the Send Command Complete transport protocol service to request that a UAS target port send a SENSE IU (see 6.2.5) on the Status pipe. As a part of USB-3 SuperSpeed operation for sending a SENSE IU, the UAS target port transmits an ERDY (see 4.4) on the Status pipe.

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

A device server calls Send Command Complete () as part of the response to receiving a SCSI Command Received () call (see 7.2.3).

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

a) all the outstanding Receive Data-Out () calls (see 7.2.8), if any, for that I_T_L nexus and Command Identifier have been responded to with Data-Out Received () calls (see 7.2.9); and
b) all the outstanding Send Data-In () calls (see 7.2.6), if any, for that I_T_L nexus and Command Identifier have been responded to with Data-In Delivered () calls (see 7.2.7).

Table 26 shows how the arguments to the Send Command Complete transport protocol service are used.

Table 26 – Send Command Complete transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: a) I indicates the initiator port to which the SENSE IU is to be sent; b) T indicates the target port used to send the SENSE IU; and c) L is implied by the Command Identifier (i.e., the TAG field in the SENSE IU) as described in 4.2.2.
Command Identifier	Indicates the value used to set the TAG field in the SENSE IU.
[Sense Data]	Indicates the information used to set the SENSE DATA field in the SENSE IU.
[Sense Data Length]	Indicates the value used to set the LENGTH field in the SENSE IU.
Status	Indicates the value used to set the STATUS field in the SENSE IU.
Service Response	Is set to COMMAND COMPLETE (see SAM-6).
[Status Qualifier]	Ignored

7.2.5 Command Complete Received transport protocol service

A UAS initiator port uses the Command Complete Received transport protocol service to notify an application client that, in response to a COMMAND IU (see 6.2.2), one of the following IUs has been received:

a) a SENSE IU (see 6.2.5) (i.e., the UAS target device has completed a command); or
b) a RESPONSE IU (see 6.2.6) (e.g., an error has been detected in the processing of a tag (see 4.2) that the UAS target device is unable to associate the IU received with any command or task management function).

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

Table 27 shows how the arguments to the Command Complete Received transport protocol service are determined.

Table 27 – Command Complete Received transport protocol service arguments

Argument	Implementation
SENSE IU received	
I_T_L nexus	I_T_L nexus, where: a) I indicates the initiator port that received the SENSE IU; b) T indicates the target port from which the SENSE IU has been received; and c) L is implied by the Command Identifier (i.e., the TAG field in the SENSE IU) as described in 4.2.2.
Command Identifier	Indicates the contents of the TAG field in the SENSE IU.
[Data-in Buffer]	Internal to the UAS initiator port
[Sense Data]	Indicates the contents of the SENSE DATA field in the SENSE IU.
[Sense Data Length]	Indicates the contents of the LENGTH field in the SENSE IU.
Status	Indicates that the contents of the STATUS field in the SENSE IU.
Service Response	Is set to COMMAND COMPLETE (see SAM-6).
[Status Qualifier]	Ignored
RESPONSE IU received	
I_T_L nexus	I_T_L nexus, where: a) I indicates the initiator port that received the RESPONSE IU; b) T indicates the target port from which the RESPONSE IU has been received; and c) L is implied by the Command Identifier (i.e., the TAG field in the RESPONSE IU) as described in 4.2.2.
Command Identifier	Indicates the contents of the TAG field in the RESPONSE IU.
[Data-in Buffer]	Ignored
[Sense Data]	Ignored
[Sense Data Length]	Ignored
Status	Ignored
Service Response	Is set to SERVICE DELIVERY OR TARGET FAILURE (see SAM-6).
[Status Qualifier]	Ignored

7.2.6 Send Data-In transport protocol service

A device server uses the Send Data-In transport protocol service to request that a UAS target port cause read data to be transferred on the Data-in pipe. As part of transferring the indicated read data, the UAS target port transmits:

a) a READ READY IU (see 6.2.3) on the Status pipe; or
b) an ERDY (see 4.4) on the Data-in pipe.

Send Data-In (IN (I_T_L Nexus, Command Identifier, Device Server Buffer, Application Client Buffer Offset, Request Byte Count))

A device server calls Send Data-In () during the processing of a command that transfers read data.

For any one instance of processing the command associated with an I_T_L nexus and Command Identifier, the device server shall not call Send Data-In() after the processing of the task associated with that command has been ended by calling:

a) Send Command Complete () (see 7.2.4) for the task associated with that I_T_L nexus and Command Identifier; or
b) Task Management Function Executed () (see 7.2.14) for a task management function that terminates one or more tasks (e.g., ABORT TASK SET), including the task associated with that I_T_L nexus and Command Identifier.

Table 28 shows how the arguments to the Send Data-In transport protocol service are used.

Table 28 – Send Data-In transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: c) I indicates the initiator port through which read data is to be transferred; d) T indicates the target port through which read data is to be transferred; and e) L is implied by the Command Identifier as described in 4.2.2.
Command Identifier	Indicates the tag (see 4.2): a) for which read data is to be transferred; and b) to which: A) a READ READY IU is to be transmitted on the Status pipe; or B) an ERDY is to be transmitted on the Data-in pipe.
Device Server Buffer	Internal to the device server
Application Client Buffer Offset	Processed as zero
Request Byte Count	Indicates the length of the read data specified by the command.

7.2.7 Data-In Delivered transport protocol service

A UAS target port uses the Data-In Delivered transport protocol service to notify a device server of the results for transferring read data in response to the processing of a Send Data-In transport protocol service (see 7.2.6).

Data-In Delivered (IN (I_T_L Nexus, Command Identifier, Delivery Result))

Table 29 shows how the arguments to the Data-In Delivered transport protocol service are determined.

Table 29 – Data-In Delivered transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: a) I indicates the initiator port through which read data has been transferred; b) T indicates the target port through which read data has been transferred; and c) L is implied by the Command Identifier as described in 4.2.2.
Command Identifier	Indicates the tag (see 4.2) for which read data has been transferred.
Delivery Result	Is set to: a) DELIVERY SUCCESSFUL (see SAM-6), if the USB IN transaction receives a USB acknowledgement as described in 7.2.16; or b) DELIVERY FAILURE (see SAM-6), if the USB IN transaction does not receive a USB acknowledgement. [a]
[a] In response to this Delivery Result, the device server shall call the Send Command Complete transport protocol service (see 7.2.4) with the Status argument set to CHECK CONDITION and the Sense Data argument containing the sense key set to ABORTED COMMAND and the additional sense code set to INITIATOR RESPONSE TIMEOUT.	

7.2.8 Receive Data-Out transport protocol service

A device server uses the Receive Data-Out transport protocol service to request that a UAS target port cause write data to be transferred on the Data-out pipe. As part of transferring the indicated write data, the UAS target port transmits:

a) a WRITE READY IU (see 6.2.4) on the Status pipe; or
b) an ERDY (see 4.4) on the Data-out pipe.

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

A device server calls Receive Data-Out () during the processing of a command that transfers write data.

For any one instance of processing the command associated with an I_T_L nexus and Command Identifier, the device server shall not call Receive Data-Out () after the processing of the task associated with that command has been ended by calling:

a) Send Command Complete () (see 7.2.4) for the task associated with that I_T_L nexus and Command Identifier; or
b) Task Management Function Executed () (see 7.2.14) for a task management function that terminates one or more tasks (e.g., ABORT TASK SET), including the task associated with that I_T_L nexus and Command Identifier.

Table 30 shows how the arguments to the Receive Data-Out transport protocol service are used.

Table 30 – Receive Data-Out transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: a) I indicates the initiator port through which write data is to be transferred; b) T indicates the target port through which write data is to be transferred; and c) L is implied by the Command Identifier as described in 4.2.2.
Command Identifier	Indicates the tag (see 4.2): a) for which write data is to be transferred; and b) to which: A) a WRITE READY IU is to be transmitted on the Status pipe; or B) an ERDY is to be transmitted on the Data-out pipe.
Application Client Buffer Offset	Processed as zero
Request Byte Count	Indicates the length of the write data specified by the command.
Device Server Buffer	Internal to the device server

7.2.9 Data-Out Received transport protocol service

A UAS target port uses the Data-out Received transport protocol service to notify a device server of the results for transferring write data in response to the processing of a Receive Data-Out transport protocol service (see 7.2.8).

Data-out Received (IN (I_T_L Nexus, Command Identifier, Delivery Result))

Table 31 shows how the arguments to the Data-out Received transport protocol service are determined.

Table 31 – Data-Out Received transport protocol service arguments

Argument	Implementation
I_T_L nexus	I_T_L nexus, where: a) I indicates the initiator port through which write data has been transferred; b) T indicates the target port through which write data has been transferred; and c) L is implied by the Command Identifier as described in 4.2.2.
Command Identifier	Indicates the tag (see 4.2) for which write data has been transferred.
Delivery Result	Is set to: a) DELIVERY SUCCESSFUL (see SAM-6), if the USB OUT transaction receives a USB acknowledgement as described in 7.2.16; or b) DELIVERY FAILURE (see SAM-6 and table 32).

For the Data-Out Received transport protocol service, the possible Delivery Result argument DELIVERY FAILURE values are shown in table 32. If the Delivery Result argument is set to one of the values shown in table 32, then the device server shall call the Send Command Complete transport protocol service (see 7.2.4) with the Status argument set to CHECK CONDITION and the Sense Data argument containing the sense key set to ABORTED COMMAND and the additional sense code set to appropriate value shown in table 32.

Table 32 – DELIVERY FAILURE to additional sense code mapping

Delivery Result	Additional sense code (see SPC-6)
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 [a]	INITIATOR RESPONSE TIMEOUT
[a] This Delivery Result indicates that the USB OUT transaction has not received a USB acknowledgement as described in 7.2.16	

7.2.10 Terminate Data Transfer transport protocol service

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

Terminate Data Transfer (IN (Nexus, Command Identifier))

Table 33 shows how the arguments to the Terminate Data Transfer transport protocol service are used.

Table 33 – Terminate Data Transfer transport protocol service arguments

Argument	Implementation
Nexus	I_T nexus or I_T_L nexus specifying the scope of the data transfers to terminate.
Command Identifier	Indicates the tag (see 4.2), if any, associated with the data transfers to terminate.

7.2.11 Data Transfer Terminated transport protocol service

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

Data Transfer Terminated (IN (Nexus, Command Identifier))

Table 34 shows how the arguments to the Data Transfer Terminated transport protocol service are determined.

Table 34 – Data Transfer Terminated transport protocol service arguments

Argument	Implementation
Nexus	I_T nexus or I_T_L nexus indicating the scope of the terminated data transfers.
Command Identifier	Indicates the tag (see 4.2), if any, associated with the terminated data transfers.

7.2.12 Send Task Management Request transport protocol service

An application client uses the Send Task Management Request transport protocol service to request that a UAS initiator port send a TASK MANAGEMENT IU (see 6.2.7) on the Command pipe.

Send Task Management Request (IN (Nexus, Management Identifier, Function Identifier, [Command Identifier]))

Table 35 shows how the arguments to the Send Task Management Request transport protocol service are used.

Table 35 – Send Task Management Request transport protocol service arguments

Argument	Implementation
Nexus	I_T nexus or I_T_L nexus specifying the nexus for the task management function as shown in table 36, where: a) I specifies the initiator port used to send the TASK MANAGEMENT IU; b) T specifies the target port to which the TASK MANAGEMENT IU is to be sent; and c) L, if required by table 36, specifies the value used to set the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU.
Management Identifier	Specifies the value used to set the TAG field in the TASK MANAGEMENT IU.
Function Identifier	Specifies the value used to set the TASK MANAGEMENT FUNCTION field (see table 20) in the TASK MANAGEMENT IU.
[Command Identifier]	If required by table 20, specifies the value used to set the TAG OF TASK TO BE MANAGED field in the TASK MANAGEMENT IU.

Table 36 shows the nexus usage by the task management functions defined by this standard.

Table 36 – UAS Task Management Functions Nexus Usage

Task Management Function	Nexus
ABORT TASK	I_T_L
ABORT TASK SET	I_T_L
CLEAR ACA	I_T_L
CLEAR TASK SET	I_T_L
I_T NEXUS RESET	I_T
LOGICAL UNIT RESET	I_T_L
QUERY TASK	I_T_L
QUERY TASK SET	I_T_L
QUERY ASYNCHRONOUS EVENT	I_T_L

7.2.13 Task Management Request Received transport protocol service

A UAS target port uses the Task Management Request Received transport protocol service to notify a task manager that a TASK MANAGEMENT IU (see 6.2.7) has been received.

Task Management Request Received (IN (Nexus, Management Identifier, Function Identifier, [Command Identifier]))

Table 37 shows how the arguments to the Task Management Request Received transport protocol service are determined.

Table 37 – Task Management Request Received transport protocol service arguments

Argument	Implementation
Nexus	I_T nexus or I_T_L nexus specifying the nexus for the task management function as shown in table 36, where: a) I specifies the initiator port from which the TASK MANAGEMENT IU was received; b) T specifies the target port that received the TASK MANAGEMENT IU; and c) L, if required by table 36, specifies the contents of the LOGICAL UNIT NUMBER field in the TASK MANAGEMENT IU.
Management Identifier	Specifies the contents of the TAG field in the TASK MANAGEMENT IU.
Function Identifier	Specifies the contents of the TASK MANAGEMENT FUNCTION field (see table 20) in the TASK MANAGEMENT IU.
[Command Identifier]	If required by table 20, specifies the contents of the TAG OF TASK TO BE MANAGED field in the TASK MANAGEMENT IU.

7.2.14 Task Management Function Executed transport protocol service

A task manager uses the Task Management Function Executed transport protocol service to request that a UAS target port send a RESPONSE IU (see 6.2.6) on the Status pipe. As a part of USB-3 SuperSpeed operation for sending a RESPONSE IU, the UAS target port transmits an ERDY (see 4.4) on the Status pipe.

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

A task manager calls Task Management Function Executed () in response to receiving a Task Management Request Received () call (see 7.2.13).

Table 38 shows how the arguments to the Task Management Function Executed transport protocol service are used.

Table 38 – Task Management Function Executed transport protocol service arguments

Argument	Implementation
Nexus	I_T nexus or I_T_L nexus indicating the nexus associated with the task management function as shown in table 36, where: a) I indicates the initiator port to which the RESPONSE IU is to be sent; b) T indicates the target port used to send the RESPONSE IU; and c) L, if required by table 36, is implied by the Management Identifier (i.e., the TAG field in the RESPONSE IU) as described in 4.2.2.
Management Identifier	Indicates the value used to set the TAG field in the RESPONSE IU.
Service Response	Indicates the value used to set the RESPONSE CODE field in the RESPONSE IU as shown in table 39.
[Additional Response Information]	If present (see SAM-6), indicates the value used to set the ADDITIONAL RESPONSE INFORMATION field in the RESPONSE IU.

The associations between Service Response argument values and the RESPONSE IU RESPONSE CODE field is shown in table 39.

Table 39 – Service Response argument associations with the RESPONSE CODE field

Service Response [a]	Associated RESPONSE CODE field (see table 18)
FUNCTION COMPLETE	TASK MANAGEMENT FUNCTION COMPLETE
FUNCTION SUCCEEDED	TASK MANAGEMENT FUNCTION SUCCEEDED
FUNCTION REJECTED	TASK MANAGEMENT FUNCTION NOT SUPPORTED
INCORRECT LOGICAL UNIT NUMBER	INCORRECT LOGICAL UNIT NUMBER
SERVICE DELIVERY OR TARGET FAILURE - Internal Failure	TASK MANAGEMENT FUNCTION FAILED
SERVICE DELIVERY OR TARGET FAILURE - Invalid IU	INVALID INFORMATION UNIT
SERVICE DELIVERY OR TARGET FAILURE - Overlapped Tag	OVERLAPPED TAG ATTEMPTED

[a] Names that are in small caps are described in SAM-6. Words that are not in small caps are associated with Service Responses that are defined in this standard.

7.2.15 Received Task Management Function Executed transport protocol service

A UAS initiator port uses the Received Task Management Function Executed transport protocol service to notify an application client that a response to a TASK MANAGEMENT IU (see 6.2.7) has been received (i.e., the UAS initiator port has received in a RESPONSE IU (see 6.2.6) with a tag (see 4.2) that is associated with the tag in a TASK MANAGEMENT IU).

If the UAS initiator port receives a RESPONSE IU with a tag that is not associated with the tag in a TASK MANAGEMENT IU, then see 7.2.5.

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

Table 40 shows how the arguments to the Received Task Management Function Executed transport protocol service are determined.

Table 40 – Received Task Management Function Executed transport protocol service arguments

Argument	Implementation
Nexus	I_T nexus or I_T_L nexus indicating the nexus associated with the task management function as shown in table 36, where: a) I indicates the initiator port that received the RESPONSE IU; b) T indicates the target port from which the RESPONSE IU was received; and c) L, if required by table 36, is implied by the Management Identifier (i.e., the contents of the TAG field in the RESPONSE IU) as described in 4.2.2.
Management Identifier	Indicates the contents of the TAG field in the RESPONSE IU.
Service Response	Indicates the contents of the RESPONSE CODE field in the RESPONSE IU as shown in table 39.
[Additional Response Information]	If present (see SAM-6), indicates the contents of the ADDITIONAL RESPONSE INFORMATION field in the RESPONSE IU.

7.2.16 USB Acknowledgement

Table 41 defines USB Acknowledgement.

Table 41 – USB Acknowledgement

USB Protocol	Acknowledgement Definition
USB OUT transactions	The transaction is acknowledged when the device transmits the 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 the 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] Receipt of the USB ACK packet for the last data packet allows the device to transmit status. USB-3 implementations retry the data transfer for any failed data packet before allowing the device to transmit status on the status endpoint as a result of USB-3 error handling rules for bulk data transfers and endpoint selection (see USB-3).

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.

Annex A
(Informative)

Bibliography

INCITS 481, *Information technology - Fibre Channel Protocol for SCSI - 4 (FCP-4)*

INCITS 484, *Information technology - SCSI Media Changer Commands - 3 (SMC-3)*

INCITS 506, *Information technology - SCSI Block Commands - 4 (SBC-4)* (under development)

INCITS 555, *Information technology - SCSI Enclosure Services - 4 (SES-4)* (under development)

INCITS 534, *Information technology - Serial Attached SCSI - 4 (SAS-4)*

INCITS 538, *Information technology - SAS Protocol Layer - 4 (SPL-4)*

Universal Serial Bus Mass Storage Class USB Attached SCSI Protocol (UASP), Revision 1.0 [1]

1. For information on the current status of USB documents, see the USB Implementors Forum at http://www.usb.org.