ISO/IEC 14776-251

Edition 1.0 2014-04

INTERNATIONAL STANDARD

colour inside

Information technology – Small computer system interface (SCSI) – Part 251: USB Attached SCSI (UAS)

ISO/IEC 14776-251:2014-04(en)

IEC Central Office
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue
The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,…). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org
The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

ISO/IEC 14776-251

Edition 1.0 2014-04

INTERNATIONAL STANDARD

colour inside

Information technology – Small computer system interface (SCSI) – Part 251: USB Attached SCSI (UAS)

INTERNATIONAL ELECTROTECHNICAL COMMISSION

PRICE CODE B

ICS 35.200

ISBN 978-2-8322-1507-4

CONTENTS

INFORMATION TECHNOLOGY – SMALL COMPUTER SYSTEM INTERFACE (SCSI) –

Part 251: USB Attached SCSI (UAS)

FOREWORD

1) ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

2) The formal decisions or agreements of IEC and ISO on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees and ISO member bodies.

3) IEC, ISO and ISO/IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees and ISO member bodies in that sense. While all reasonable efforts are made to ensure that the technical content of IEC, ISO and ISO/IEC publications is accurate, IEC or ISO cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.

4) In order to promote international uniformity, IEC National Committees and ISO member bodies undertake to apply IEC, ISO and ISO/IEC publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any ISO, IEC or ISO/IEC publication and the corresponding national or regional publication should be clearly indicated in the latter.

5) ISO and IEC do not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. ISO or IEC are not responsible for any services carried out by independent certification bodies.

6) All users should ensure that they have the latest edition of this publication.

7) No liability shall attach to IEC or ISO or its directors, employees, servants or agents including individual experts and members of their technical committees and IEC National Committees or ISO member bodies for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication of, use of, or reliance upon, this ISO/IEC Publication or any other IEC, ISO or ISO/IEC publications.

8) Attention is drawn to the normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.

9) Attention is drawn to the possibility that some of the elements of this ISO/IEC Publication may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

International Standard ISO/IEC 14776-251 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

The list of all currently available parts of the ISO/IEC 14776 series, under the general title *Information technology – Small computer system interface (SCSI)*, can be found on the IEC web site.

This International Standard has been approved by vote of the member bodies, and the voting results may be obtained from the address given on the second title page.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IMPORTANT – The "colour inside" logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this publication using a colour printer.

INTRODUCTION

This International Standard standard encompasses the following:

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.
Clause 8 describes device server error handling.

SCSI standards family

Figure 1 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.

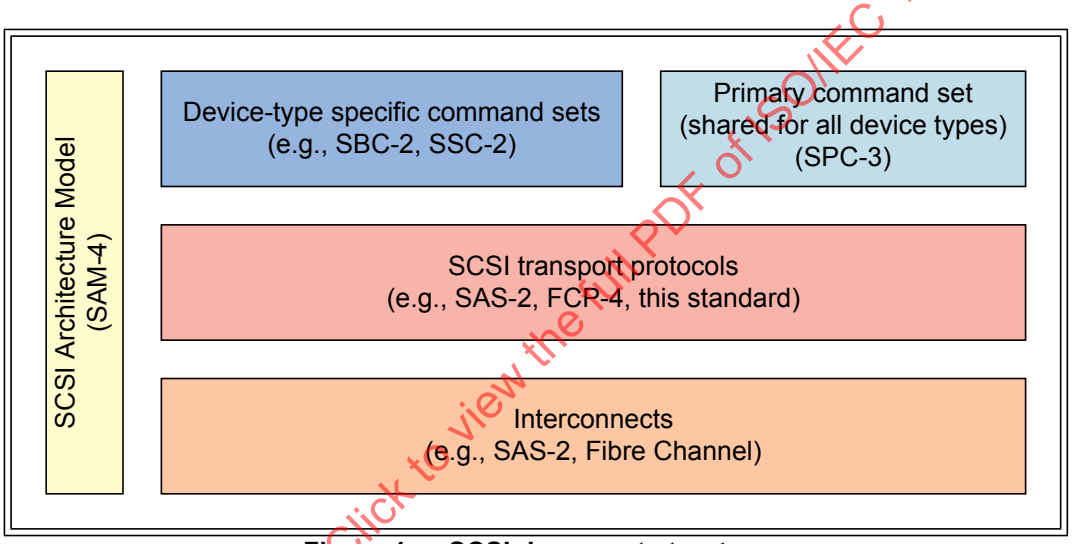


Figure 1 — SCSI document structure

The SCSI document structure in figure 1 is intended to show the general applicability of the documents to one another. Figure 1 is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture.

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 shared by all SCSI devices.

Shared 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 introduction.

INFORMATION TECHNOLOGY – SMALL COMPUTER SYSTEM INTERFACE (SCSI) –

Part 251: USB Attached SCSI (UAS)

1 Scope

This part of ISO/IEC 14776 describes a SCSI transport protocol (see ISO/IEC 14776-414) for USB-2 and USB-3 with the following properties:

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

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 14776-414, *Information technology, Small Computer System Interface (SCSI) – Part 414: SCSI Architecture Model-4* (herein referred to as SAM-4) [1]

IEC 62680-1, *Universal serial bus interfaces for data and power – Part 1: Universal serial bus specification, revision 2.0* (herein referred to as USB-2)

ANSI INCITS 513-2004, *SCSI Primary Commands-4* (herein referred to as SPC-4) [T10/1731-D][2]

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

Universal Serial Bus Mass Storage Class Specification Overview Rev 1.3 (herein referred to as MSC). September 5, 2008[3]

3 Terms, definitions, symbols, abbreviations and conventions

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
application client
object that is the source of SCSI commands

Note 1 to entry: See ISO/IEC 14776-414.

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 1

1. ANSI INCITS 447-2008

2. planned as ISO/IEC 14776-454

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

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

3.1.3
Bulk-in pipe
pipe (see 3.1.9) 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 0

3.1.5
Bulk-out pipe
pipe (see 3.1.9) used to transfer data and commands from the UAS initiator port to the UAS target port

3.1.6
default pipe
message pipe (see 3.1.9) created by the USB System Software to pass control and status information between the host and a USB device's endpoint zero (see USB-2)

3.1.7
Information Unit
IU
formatted collection of data that carries command, task management function, sense, response, read ready, or write ready information (see 6.2)

3.1.8
Logical Unit Number
LUN
64-bit identifier for a logical unit

Note 1 to entry: See ISO/IEC 14776-414.

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, as requested by the Send Data-In transport protocol service

3.1.11
service deliver subsystem
part of the USB I/O system which transmits information between the UAS initiator port and the UAS target port

EXAMPLE USB hubs, USB cables.

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

Note 1 to entry: See ISO/IEC 14776-414.

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

Note 1 to entry: See Universal Serial Bus 3.0.

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

3.1.15
UAS initiator device
USB host that contains one or more UAS initiator ports

Note 1 to entry: For USB host, see Universal Serial Bus 2.0 and Universal Serial Bus 3.0.

3.1.16
UAS initiator port
USB host and USB host port components

Note 1 to entry: For USB host, see Universal Serial Bus 2.0.

Note 2 to entry: For USB host port components, see Universal Serial Bus 2.0 and Universal Serial Bus 3.0.

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

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

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

Note 1 to entry: See Universal Serial Bus 2.0 and Universal Serial Bus 3.0.

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

Note 1 to entry: See 3.1.9, and Universal Serial Bus 2.0 and Universal Serial Bus 3.0.

3.1.21
USB interface
description of one or more USB endpoints

Note 1 to entry: See Universal Serial Bus 2.0 and Universal Serial Bus 3.0.

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, as requested by the Request Data-Out transport protocol service

3.2 Symbols and abbreviations

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

3.3 Keywords

invalid

a keyword used to describe an illegal or unsupported bit, byte, word, field or code value; receipt by a device server of an invalid bit, byte, word, field or code value shall be reported as error

mandatory

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

may

a keyword that indicates flexibility of choice with no implied preference

may not

a keyword that indicates flexibility of choice with no implied preference

obsolete

a keyword indicating that an item was defined in prior SCSI standards but has been removed from this standard

option, optional

keywords that describe features that are not required to be implemented by this standard; however, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard

reserved

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

shall

a keyword indicating a mandatory requirement; designers are required to implement all such mandatory requirements to ensure interoperability with other products that conform to this standard

should

a keyword indicating flexibility of choice with a strongly preferred alternative

vendor specific

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

3.4 Editorial conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 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 lowercase or uppercase letters show no ordering relationship between the listed items.

EXAMPLE 1 - The following list shows no relationship between the named items:

a) red (i.e., one of the following colors):
   A) crimson: or
   B) amber;
b) blue; or
c) green.

Lists sequenced by numbers show an ordering relationship between the listed items.

EXAMPLE 2 - The following list shows an ordered relationship between the named items:

1) top;
2) middle; and
3) bottom.

Lists are associated with an introductory paragraph or phrase, and are numbered relative to that paragraph or phrase (i.e., all lists begin with an a) or 1) entry).

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 2 is an example sequence figure. A line with an arrowhead that points to the responder represents a communication from the requestor to the responder. A line with an arrowhead that points to the requestor represents a communication from the responder to the requestor.

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

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

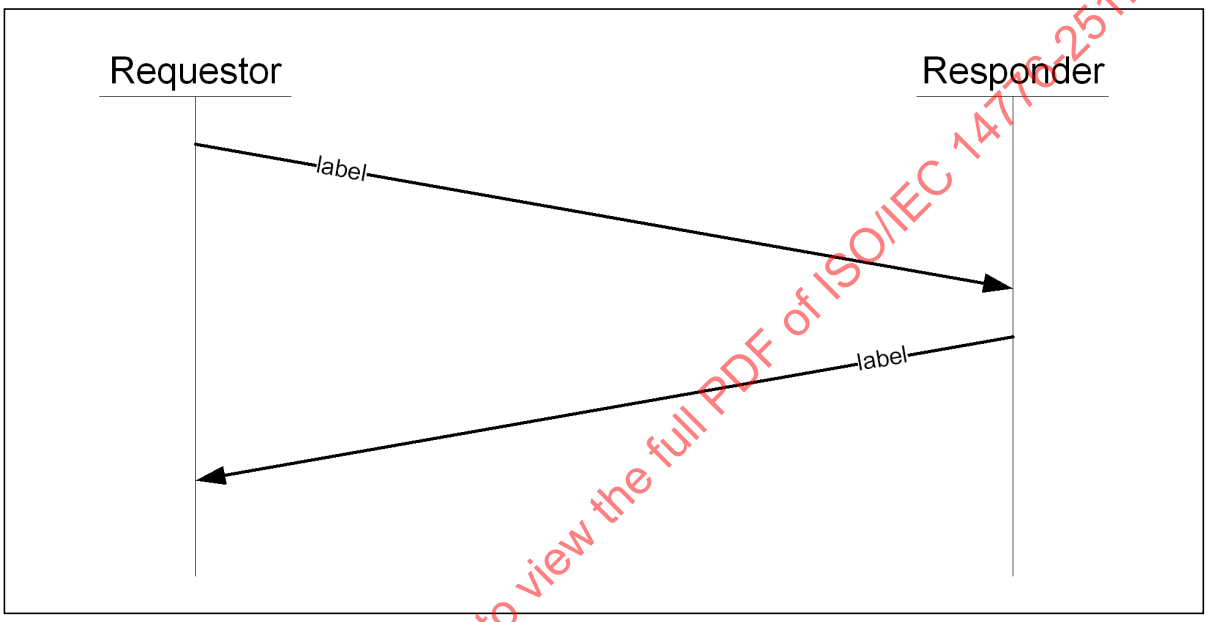


Figure 2 – Example Sequence figure

3.7 Notation for procedures and functions

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

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

Where:

Result	A single value representing the outcome of the procedure or function.
Procedure Name	descriptive name for the function to be performed.
IN (Input-1, Input-2, …)	A comma-separated list of names identifying caller-supplied input data objects.
OUT (Output-1, Output-2, …)	A comma-separated list of names identifying output data objects to be returned by the procedure.
[…]	Brackets enclose optional or conditional parameters and arguments.

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

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

4 Model

4.1 Overview

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

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

   and

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

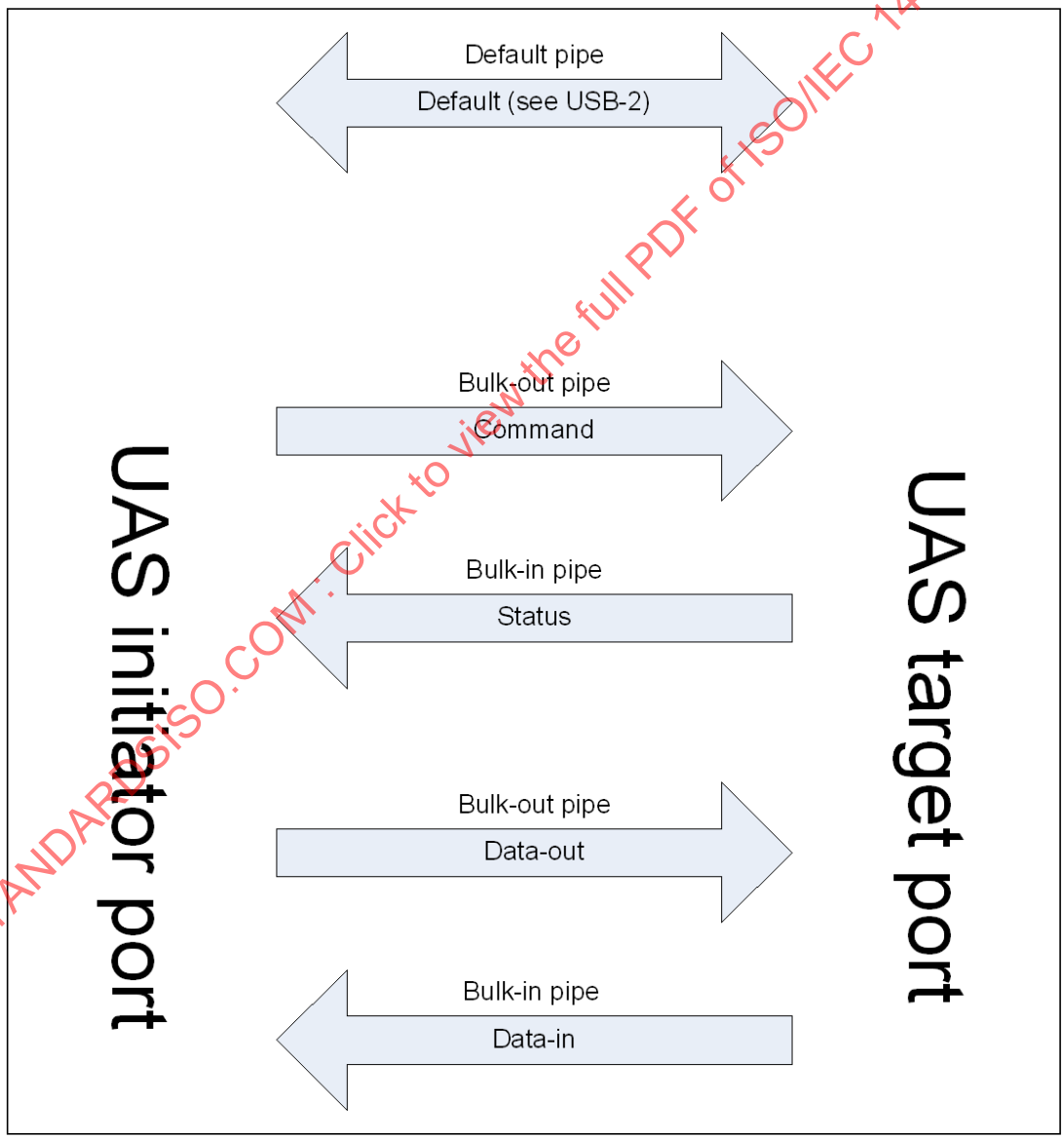


Figure 3 – USB Model

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

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

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

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

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

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

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

4.2 Tag handling

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

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

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

If a UAS target device performs tag checking and:

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

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

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

4.3 Data transfers

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

a) transfer data;
b) accept COMMAND IUs and TASK MANAGEMENT IUs using the Command pipe;
c) process the commands and task management functions; and
d) return status for commands and task management functions on the Status pipe.

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

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

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

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

4.4 UAS domain

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

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

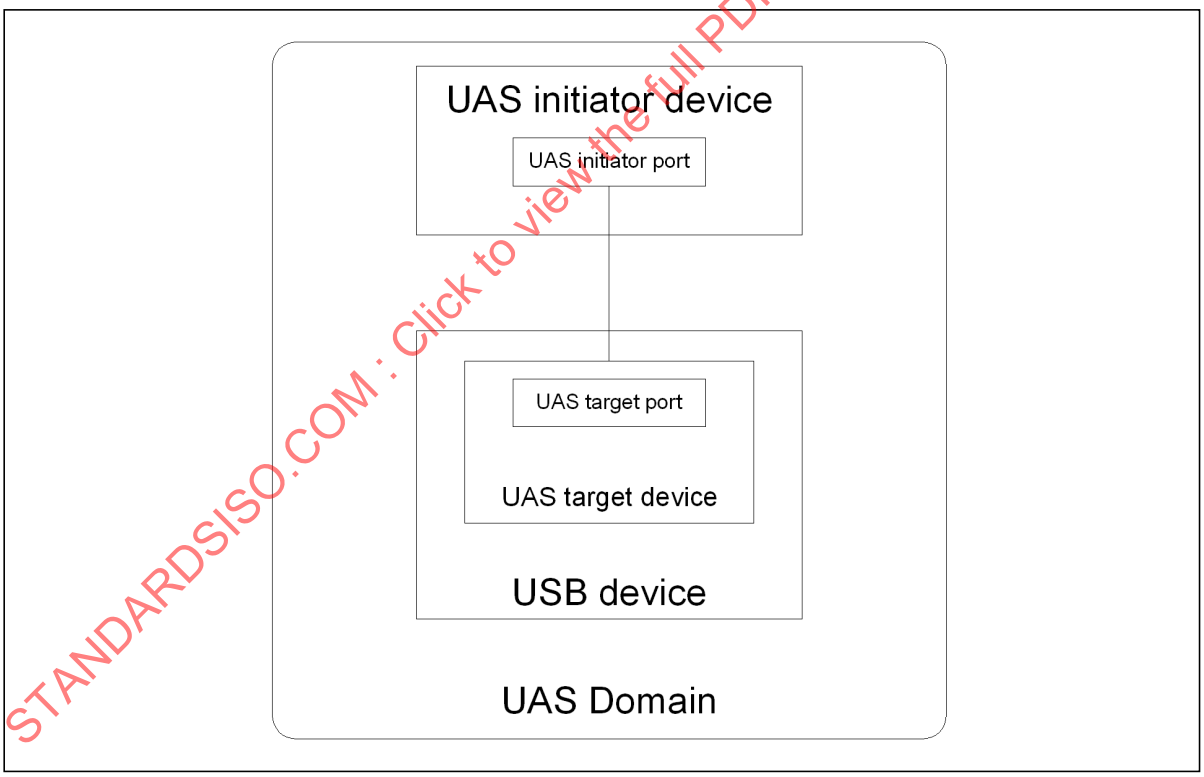


Figure 4 – Example Simple UAS domain

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

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

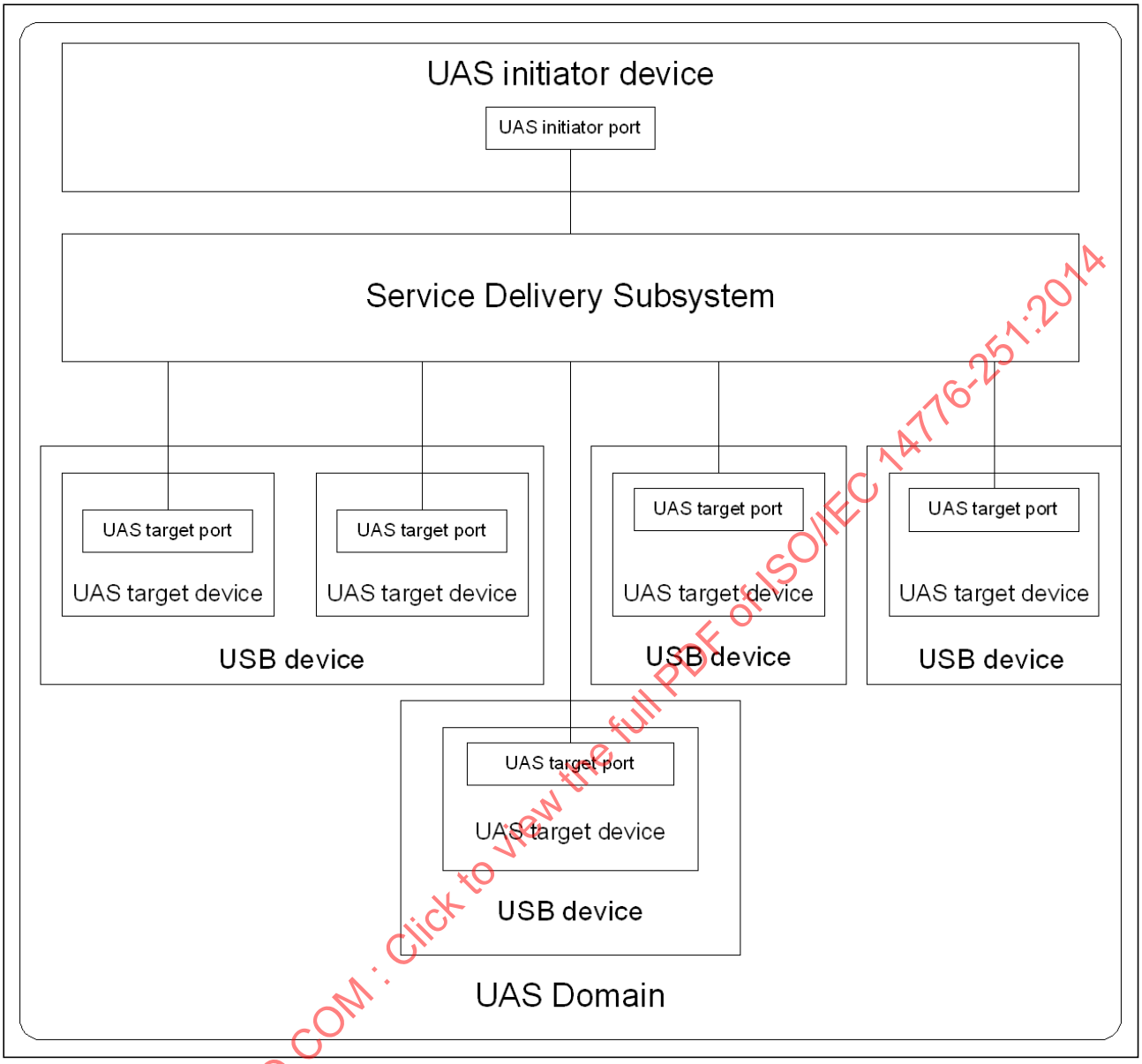


Figure 5 – Example Complex UAS Domain

4.5 Addressing

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

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

4.6 World wide name

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

4.7 Resets

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

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

4.8 I_T Nexus loss

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

4.9 Target power loss expected

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

4.10 USB error handling

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

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

5 USB

5.1 Overview

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

5.2 USB resource requirements

5.2.1 Overview

This standard requires a minimum of:

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

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

5.2.2 USB class specific requests

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

5.2.3 USB descriptors

5.2.3.1 Device descriptor

Table 2 describes the Device descriptor format.

Table 2 – 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 2 (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 3 describes the Configuration descriptor format.

Table 3 – Configuration descriptor

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

The:

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

shall be set to the value defined in table 3 (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 4 describes the Interface descriptor format.

Table 4 – Interface Descriptor

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

The:

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

shall be set to the value defined in table 4 (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 5 describes the Bulk-in endpoint descriptor format.

Table 5 – Bulk-in endpoint descriptor

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

The:

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

shall be set to the value defined in table 5 (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 6 describes the Bulk-out endpoint descriptor format.

Table 6 – Bulk-out endpoint descriptor

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

The:

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

shall be set to the value defined in table 5 (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 7 describes the format of the Pipe Usage Class specific descriptor.

Table 7 – Pipe Usage Descriptor

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

The B LENGTH field shall be set to the value defined in table 7.

The B DESCRIPTOR TYPE field shall be set to the value defined in table 7 (see MSC).

The B PIPE ID field identifies the pipe associated with the endpoint descriptor (see table 8).

Table 8 – Pipe ID

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

6 Transport

6.1 Overview

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

6.2 IUs

6.2.1 Overview

Table 9 is a summary of the Information Units (IUs) and the associated IU ID field.

Table 9 – IU ID field summary

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

All IUs include the header defined in table 10 as the first bytes of the IU.

Table 10 – IU Header

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

The IU ID field identifies the type of IU (see table 9).

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 return a RESPONSE IU with the RESPONSE CODE field set to INVALID INFORMATION UNIT.

6.2.2 COMMAND IU

The COMMAND IU shall be contained in a single USB packet and shall not share a USB packet with any other IU. Table 11 defines the COMMAND IU.

Table 11 – COMMAND IU

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

The IU ID field shall be set to the value defined in table 11.

The TAG field is described in 4.2.

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

Table 12 defines the TASK ATTRIBUTE field.

Table 12 – TASK ATTRIBUTE field

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

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

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

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

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

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

6.2.3 READ READY IU

The READ READY IU is sent by a UAS target port to inform the UAS initiator port that the UAS target port is ready to send data for a data-in command (see 6.3.5) or a bi-directional command (see 6.3.7). UAS target devices that connect via SuperSpeed (see USB-3) shall return an ERDY transaction packet (see USB-3) on the Data-in pipe using the tag as the stream ID instead of the READ READY IU on the Status pipe. The READ READY IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 13 describes the READ READY IU.

TABLE 13 – 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 13.

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

6.2.4 WRITE READY IU

The WRITE READY IU is sent by a UAS target port to request write data from the UAS initiator port during a data-out command (see 6.3.4) or a bi-directional command (see 6.3.7). UAS target devices that connect via SuperSpeed (see USB-3) shall return an ERDY transaction packet (see USB-3) on the Data-out pipe using the tag as the stream ID instead of the WRITE READY IU on the Status pipe. The WRITE READY IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 14 defines the WRITE READY IU.

TABLE 14 – 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 14.

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 15 defines the SENSE IU.

Table 15 – SENSE IU

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

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

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

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-4).

6.2.6 RESPONSE IU

The RESPONSE IU is used to pass task management status information from the UAS target port to the UAS initiator port. The RESPONSE IU may be returned in response to a COMMAND IU as a means to report an error condition detected by the transport. Each RESPONSE IU shall be sent in a single USB packet. The RESPONSE IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 16 defines the RESPONSE IU.

Table 16 – RESPONSE IU

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

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

The RESPONSE CODE field (see table 17) indicates the status of a task management function.

Table 17 – RESPONSE CODE **field**

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

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

6.2.7 TASK MANAGEMENT IU

The TASK MANAGEMENT IU is sent by a UAS initiator port to request that a task management function be processed by the task manager in a logical unit. The TASK MANAGEMENT IU shall be contained in a single USB packet and shall not share a USB packet with any other Information Unit. Table 18 defines the TASK MANAGEMENT IU format.

Table 18 – TASK MANAGEMENT IU

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

The IU ID field shall be set to the value defined in table 18.

The TAG field is described in 4.2.

Table 19 defines the TASK MANAGEMENT FUNCTION field.

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

Table 19 – TASK MANAGEMENT FUNCTION field (part 1 of 2)

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

[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.

Table 19 – TASK MANAGEMENT FUNCTION field (part 2 of 2)

Code	Task management function	Uses the		Description
		LUN [a]	TOTTBM [b]	
41h-7Fh	Reserved			
80h	QUERY TASK	yes	yes	The task manager shall perform the QUERY TASK task management function using the value of the LOGICAL UNIT NUMBER field and the value of the TAG OF TASK TO BE MANAGED field to determine the task to be queried (see SAM-4). [c]
81h	QUERY TASK SET	yes	no	The task manager shall perform the QUERY TASK SET task management function using the value of the LOGICAL UNIT NUMBER field to determine the task set to be queried (see SAM-4). [c]
82h	QUERY ASYNCHRONOUS EVENT	yes	no	The task manager shall perform the QUERY ASYNCHRONOUS EVENT task management function using the value of the LOGICAL UNIT NUMBER field (see SAM-4). [c]
83h-FFh	Reserved			

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

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

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

6.3 Information unit sequences

6.3.1 Overview

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

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

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

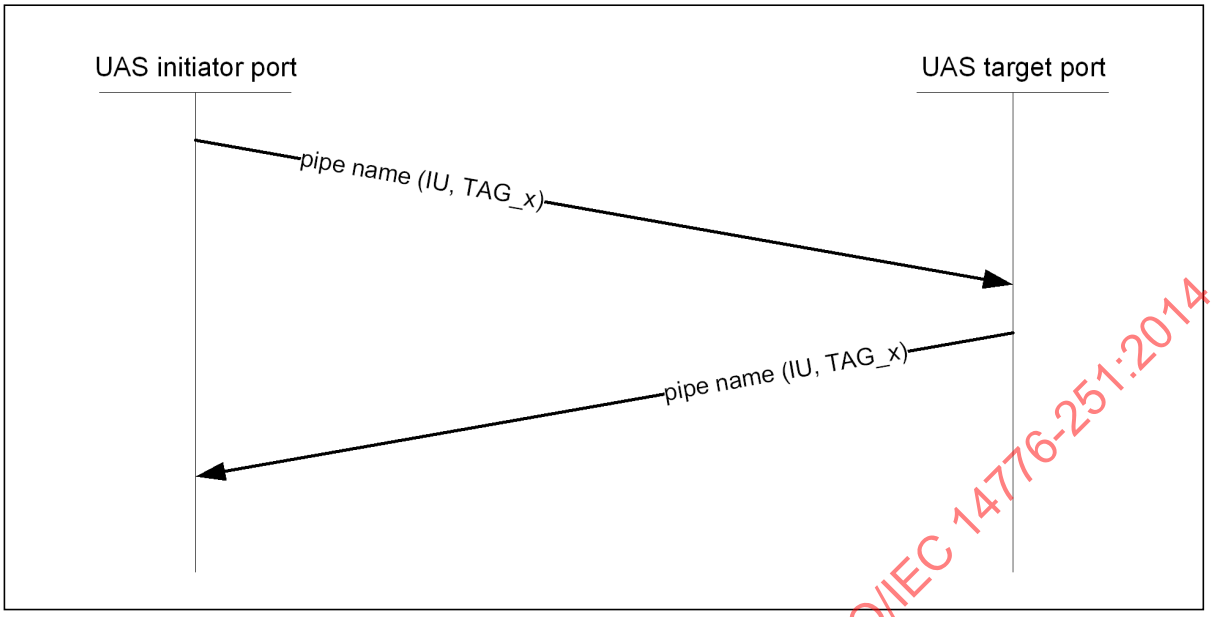


Figure 6 – UAS sequence figure notation

6.3.2 Non-data command/sense sequence

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

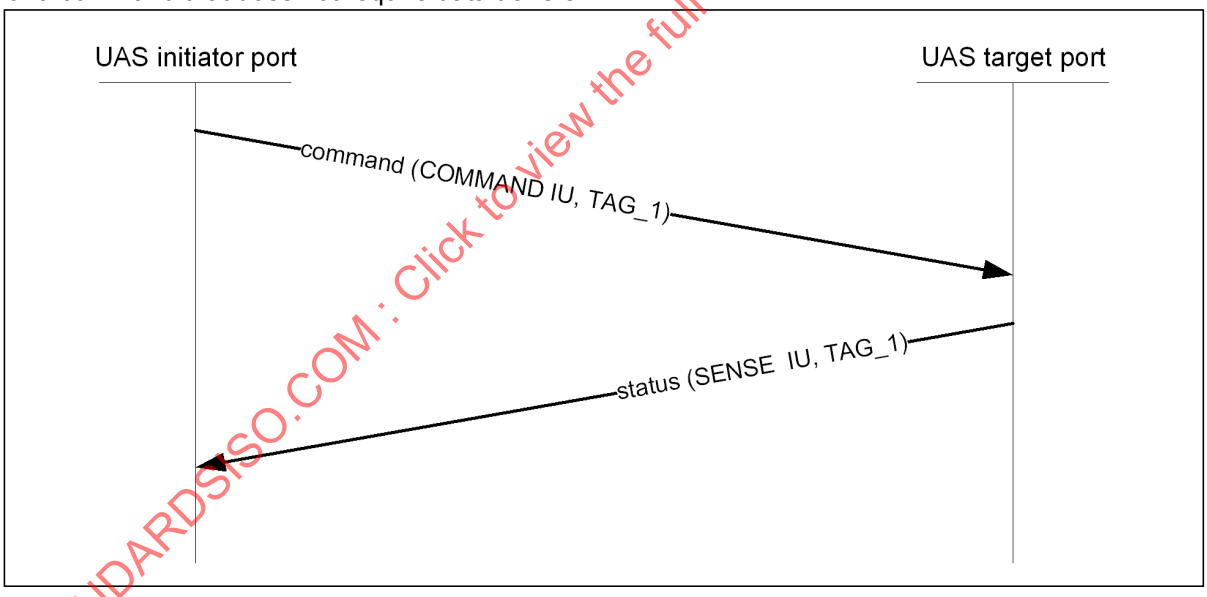


Figure 7 – Non-data transfer with Sense

6.3.3 Non-data command/response sequence

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

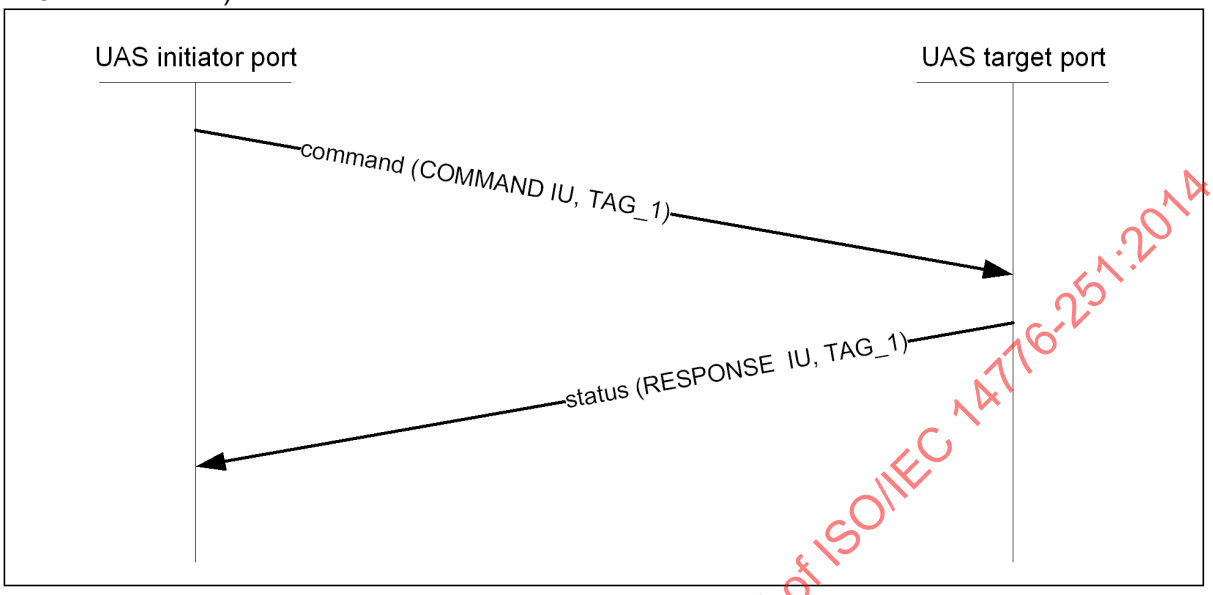
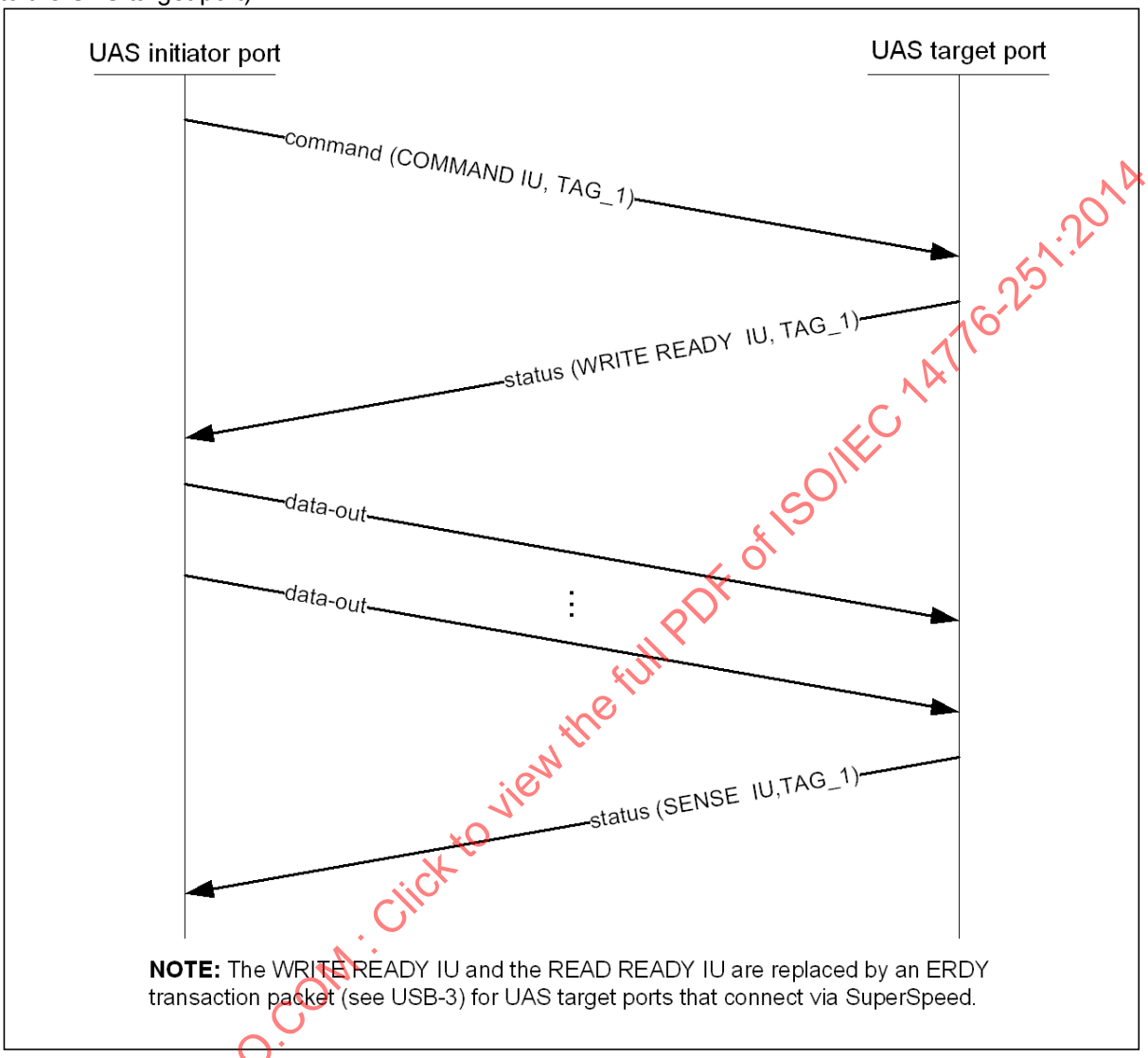


Figure 8 – Non-data Transfer with Response

6.3.4 Data-out command sequence

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



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

Figure 9 – Write Data Transfer

6.3.5 Data-in command sequence

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

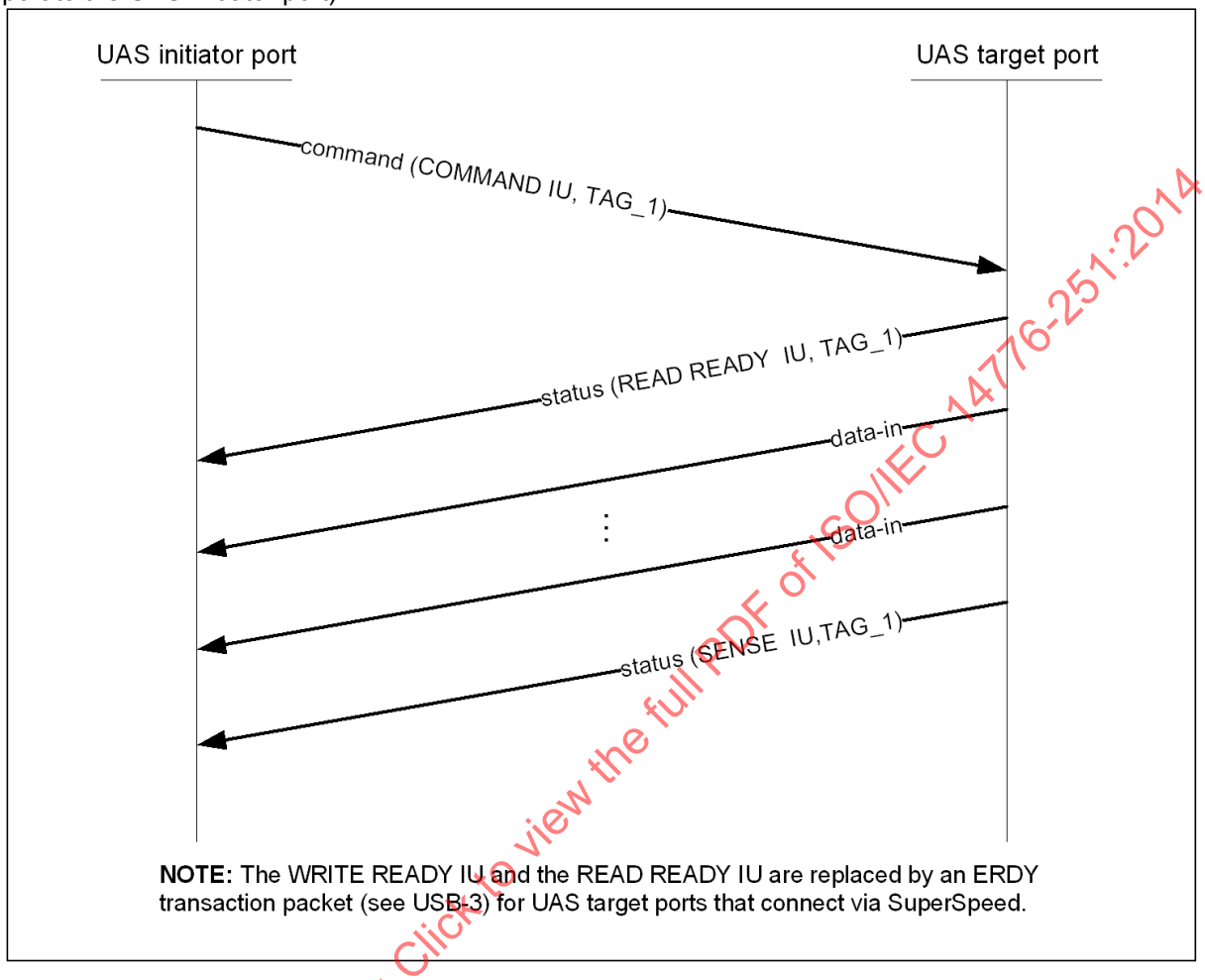


Figure 10 – Read Data Transfer

6.3.6 Task management function sequence

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

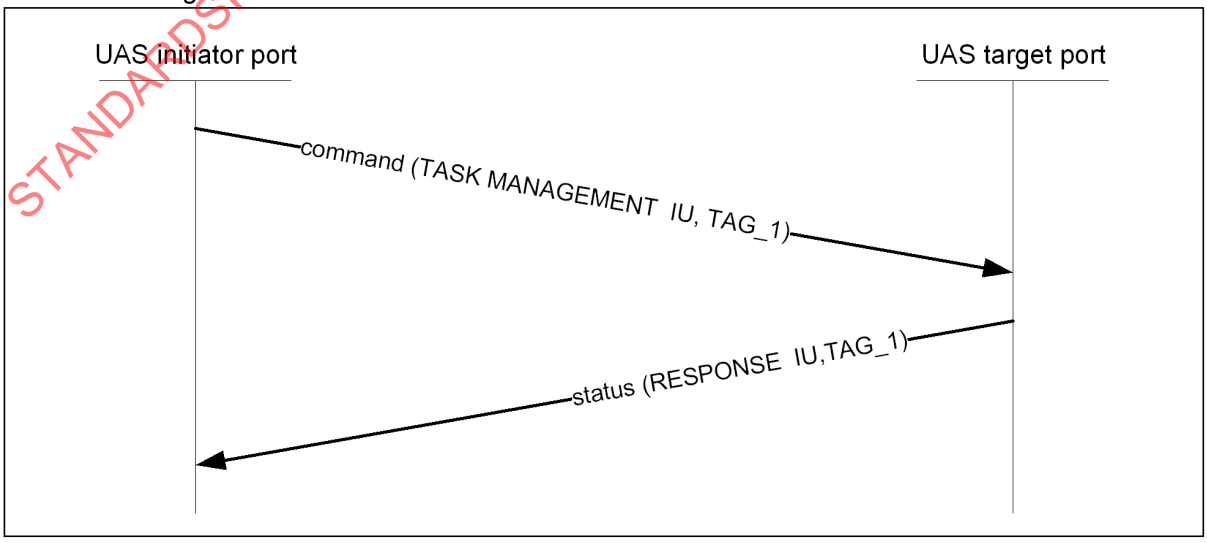


Figure 11 – Task management

6.3.7 Bi-directional command sequence

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

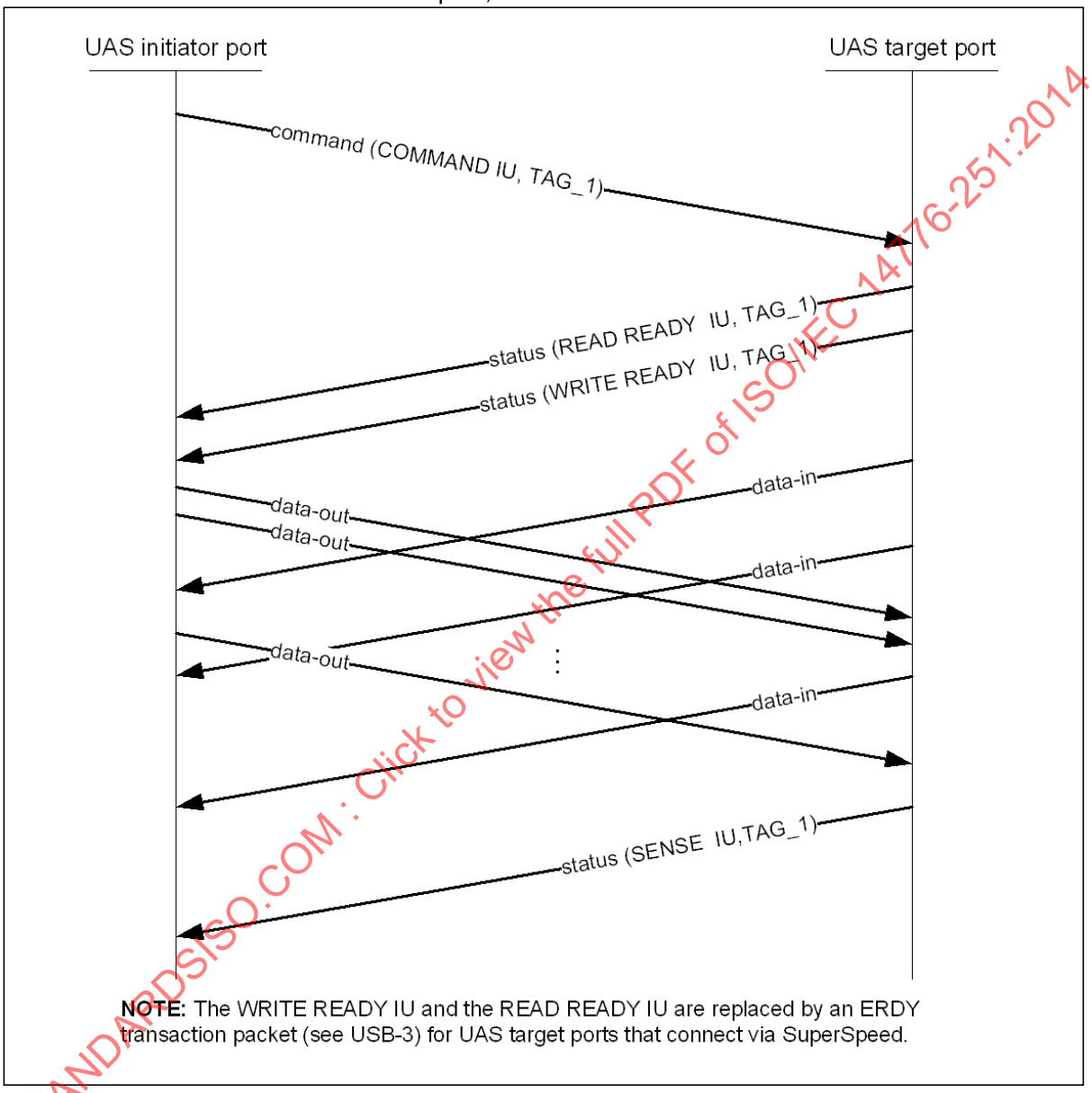


Figure 12 – Bi-directional Data Transfer

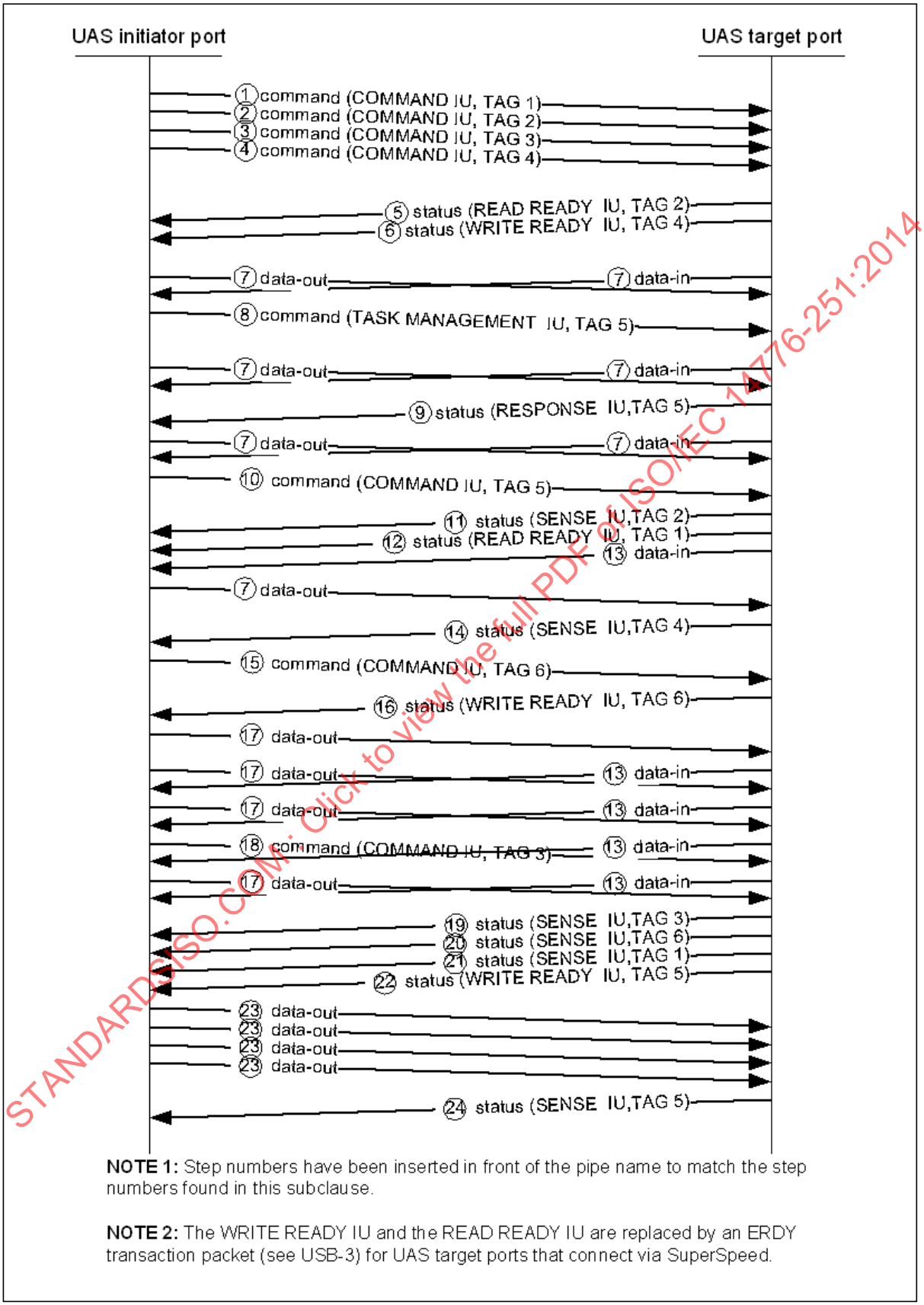
NOTE 1 - The order of the data-in and data-out phases of a bidirectional command may be influenced by both the definition of the bidirectional command and the capabilities of the USB target.

6.3.8 Multiple command example

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

1) the UAS initiator port transfers a read command with TAG 1;
2) the UAS initiator port transfers a read command with TAG 2;
3) the UAS initiator port transfers a write command with TAG 3;
4) the UAS initiator port transfers a write command with TAG 4;
5) the UAS target port requests to transfer the read data for TAG 2;

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



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

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

Figure 13 – Multiple Command example

6.4 Transport requirements

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

7 SCSI Application Layer transport protocol services

7.1 SCSI transport protocol services overview

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

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

This standard defines the transport protocol services required by SAM-4 in support of these procedure calls. Table 20 describes the mapping of the Execute Command procedure call to transport protocol services and the UAS implementation of each transport protocol service.

Table 20 – Execute Command procedure call transport protocol services

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

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

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

a) Service Response = ABORT TASK (IN (Nexus));
b) Service Response = ABORT TASK SET (IN (Nexus));
c) Service Response = CLEAR ACA (IN (Nexus));
d) Service Response = CLEAR TASK SET (IN (Nexus));
e) Service Response = I_T NEXUS RESET (IN (Nexus));
f) Service Response = LOGICAL UNIT RESET (IN (Nexus));
g) Service Response = QUERY TASK (IN (Nexus));
h) Service Response = QUERY TASK SET (IN (Nexus)); and
i) Service Response = QUERY ASYNCHRONOUS EVENT (IN (Nexus), OUT ([Additional Response Information])).

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

Table 21 – Execute Command procedure call transport protocol services

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

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

7.2 Send SCSI Command transport protocol service

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

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

Table 22 shows how the arguments to the Send SCSI Command transport protocol service are used.

Table 22 – Send SCSI Command transport protocol service arguments

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

7.3 SCSI Command Received transport protocol service

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

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

Table 23 shows how the arguments to the SCSI Command Received transport protocol service are determined.

Table 23 – SCSI Command Received transport protocol service arguments

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

7.4 Send Command Complete transport protocol service

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

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

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

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

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