Working Draft American National Standard

Project T10/BSR INCITS 536

May 8, 2014

Information technology - Zoned Block Commands (ZBC)

T10 Technical Editor:

Your Name
Your Company Name
Address Line 1
Address Line 2
Address Line 3
USA

Telephone: xxx-xxx-xxxx
Facsimile: xxx-xxx-xxxx
Email: xxxxxx@xxxxxx

Reference number
ISO/IEC 14776-345:201x
BSR INCITS 536-201x

Points of Contact

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

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

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

T10 Vice-Chair
Mark S. Evans
Western Digital Corporation
5863 Rue Ferrari
San Jose, CA 95138
USA

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

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

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

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

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

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

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

American National Standard
for Information Technology

INCITS 536 (ZBC)

Secretariat
Information Technology Industry Council

Approved mm.dd.yy

American National Standards Institute, Inc.

ABSTRACT

This standard specifies something of interest, and here is where you give a brief description of that. This abstract should be confined to one or two short paragraphs.

The patent statement goes here.

If no patents have been disclosed place the statement in 5.5.2 shall be used.

If any patents have been disclosed place the statement in 5.5.3 shall be used.

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

Revision Information

R.1 Revision 0 (5-Dec-2013)

a) First draft. Taken from approved proposal 14-005r1.

R.2 Revision 1 (8-May-2014)

a) Incorporated 14-125r2 as approved by the May T10 plenary (minutes in 14-119), including:
   A) the addition of the Note to Entry paragraph tag from 14-006r1 T10 Style Guide for glossary/keyword entries; and
   B) application of the Note to Entry paragraph tag to entries in the keywords subclause.
b) Removed all editor's notes from 14-125r2 that apply to SPC-5 or SBC-4.
c) As per the T10 plenary motion that “the Host Managed Zoned Device column of the *Characteristics For Zoned Block Devices* table be modified so that 'Optional' is replaced by 'Not supported' for Sequential Write Preferred zones”, modify the contents of table 6 (see 14-125r2).

Because all the technical changes incorporated in revision 01 represent totally new subclauses, no changes are marked with change bars or edits tracking.

Contents

Tables

Figures

FOREWORD (This foreword is not part of this standard)

This purpose of this standard is to define the model and command set extensions to be used in conjunction with the SCSI Primary Command Set standard – 4 (SPC-4) and the SCSI Block Commands - 3 (SBC-3) to facilitate operation of SCSI zoned block devices.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, International Committee for Information Technology Standards, Information Technology Industry Council, Suite 610, 1101 K Street, NW, 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 it approved this standard, INCITS had the following members:

INCITS Technical Committee T10 on SCSI Storage Interfaces, which developed and reviewed this standard, had the following members:

John B. Lohmeyer, Chair
William Martin, Vice-Chair
Ralph O. Weber, Secretary

INTRODUCTION

General

The INCITS 536 (ZBC) standard is divided into the following clauses and annexes:

Clause 1 is the scope.
Clause 2 enumerates the normative references that apply to this standard.
Clause 3 describes the definitions, symbols, and abbreviations used in this standard.
Clause 4 describes the operational model.
Clause 5 describes commands for zoned block devices.
Clause 6 describes parameters for zoned block devices
Annex A describes example targets that implement 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.

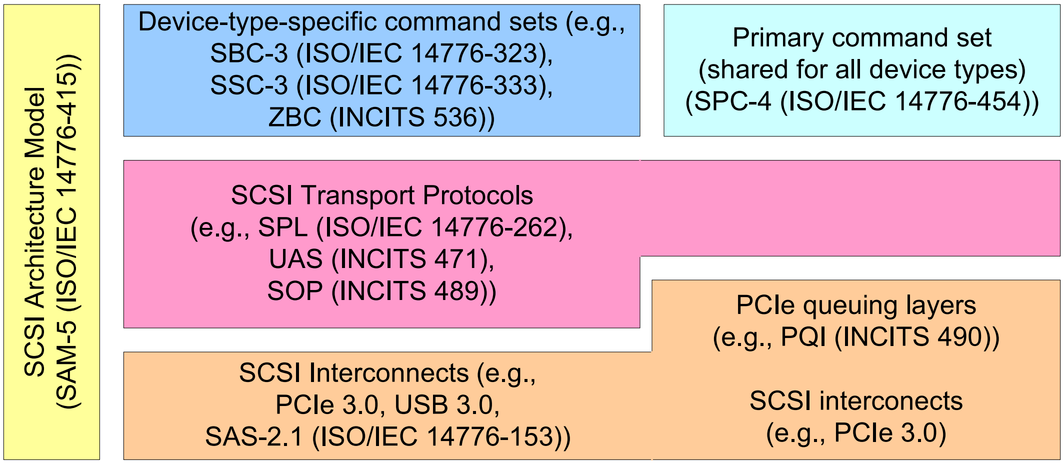

SCSI Architecture Model
(SAM-5 (ISO/IEC 14776-415))
Device-type-specific command sets (e.g.,
SBC-3 (ISO/IEC 14776-323),
SSC-3 (ISO/IEC 14776-333),
ZBC (INCITS 536))
Primary command set
(shared for all device types)
(SPC-4 (ISO/IEC 14776-454))
SCSI Transport Protocols
(e.g., SPL (ISO/IEC 14776-262),
UAS (INCITS 471),
SOP (INCITS 489))
PCIe queuing layers
(e.g., PQI (INCITS 490))
SCSI Interconnects (e.g.,
PCIe 3.0, USB 3.0,
SAS-2.1 (ISO/IEC 14776-153))
SCSI interconects
(e.g., PCIe 3.0)

Figure 0 — SCSI document structure

The roadmap 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 set of SCSI standards specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming SCSI implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability. See SAM-5 for more information about the relationships between the SCSI standards.

1 Scope

This standard defines the command set extensions to facilitate operation of SCSI zoned block devices.

The clauses in this standard, implemented in conjunction with the applicable clauses of SPC-4 and SBC-3, specify the standard command set for SCSI zoned block devices.

The objectives of this standard are to:

a) permit an application client to communicate over a SCSI service delivery subsystem (see SAM-5) with a logical unit that declares itself to be a zoned block device in the PERIPHERAL DEVICE TYPE field of the standard INQUIRY data (see SPC-4); and
b) define commands and parameters unique to the zoned block device type.

2 Normative references

2.1 Normative references

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

Copies of the following documents may be obtained from ANSI: approved ANSI standards, approved and draft international and regional standards (ISO, IEC, CEN/CENELEC, ITUT), and approved and draft foreign standards (including BSI, JIS, and DIN). For further information, contact ANSI Customer Service Department at 212-642-4900 (phone), 212-302-1286 (fax) or via the World Wide Web at http://www.ansi.org.

Additional availability contact information is provided below as needed.

2.2 Approved references

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

ISO/IEC 14776-xxx, *SCSI Primary Commands - 3 (SPC-3)* [ANSI INCITS 408-2005]

ISO/IEC 14776-xxx, *SCSI Block Commands - 2 (SBC-2)* [ANSI INCITS 405-2005]

2.3 References under development

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

ISO/IEC 14776-xxx, *SCSI Architecture Model - 5 (SAM-5)* [ANSI INCITS 515]

ISO/IEC 14776-xxx, *SCSI Primary Commands - 4 (SPC-4)* [ANSI INCITS 408]

ISO/IEC 14776-xxx, *SCSI Primary Commands - 5 (SPC-5)* [ANSI INCITS 502]

ISO/IEC 14776-xxx, *SCSI Block Commands - 3 (SBC-3)* [INCITS 514]

ISO/IEC 14776-xxx, *SCSI Block Commands - 4 (SBC-4)* [INCITS 506]

2.4 Other references

For information on the current status of the listed document(s), or regarding availability, contact the indicated organization.

SFF-8451, *SCA-2 Unshielded Connections*

NOTE 1 - For more information on the current status of the document, contact the SFF committee at 408-867-6630 (phone), or 408-867-2115 (fax). To obtain copies of this document, contact the SFF committee at 14426 Black Walnut Court, Saratoga, CA 95070 at 408-867-6630 (phone) or 408-741-1600 (fax).

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 byte
8-bit construct

3.1.2 conventional zone
range of LBAs that is not associated with a write pointer

Note 1 to entry: See 4.3.2.

3.1.3 ending LBA
highest numbered LBA of a read command or a write command

3.1.4 first LBA
lowest numbered LBA of a read operation or write operation

highest LBA

highest numbered LBA of a range of LBAs

Note 1 to entry: See 4.3.

3.1.5 host aware zoned block device
SCSI target device with the PERIPHERAL DEVICE TYPE field set to 00h (i.e., direct access block device) and the HAW_ZBC bit set to one in the Block Device Characteristics VPD page (see SBC-4)

Note 1 to entry: See 4.2.2.

3.1.6 host managed zoned block device
SCSI target device with the PERIPHERAL DEVICE TYPE field set to 14h (i.e., host managed zoned block device)

Note 1 to entry: See 4.2.3.

3.1.7 last LBA
highest numbered LBA of a read operation or write operation

3.1.8 logical block address (LBA)
value used to reference a logical block

Note 1 to entry: See SBC-4.

3.1.9 lowest LBA
lowest numbered LBA of a range of LBAs

Note 1 to entry: See 4.3.

3.1.10 read operation
process by which a device server performs operations (e.g., read cache operations (see SBC-4) and read medium operations (see SBC-4)) as described in SBC-4

Note 1 to entry: See SBC-4.

3.1.11 reset write pointer operation
process by which a device server sets a write pointer to the lowest LBA of a write pointer zone

Note 1 to entry: See 4.3.3.

3.1.12 sequential write preferred zone

write pointer zone in which the device server allows write operations that specify a first LBA within a zone that is not equal to that zone's write pointer

Note 1 to entry: See 4.3.3.2.

3.1.13 sequential write required zone

write pointer zone in which the device server requires that write operations specify a first LBA that is equal to that zone's write pointer

Note 1 to entry: See 4.3.3.3.

3.1.14 starting LBA

lowest numbered LBA of a read command or a write command

3.1.15 write operation

process by which a device server performs operations (e.g., write cache operations (see SBC-4) and write medium operations (see SBC-4)) as described in SBC-4

Note 1 to entry: See SBC-4.

3.1.16 write pointer

pointer to a logical block (i.e., an LBA) in a write pointer zone

Note 1 to entry: See 4.3.3.1.

3.1.17 write pointer zone

range of LBAs that is associated with a write pointer

Note 1 to entry: See 4.3.3.

3.1.18 zone

contiguous range of LBAs

Note 1 to entry: See 4.3.

3.1.19 zoned block device

either a host aware zoned block device or a host managed zoned block device

Note 1 to entry: See 4.2.

3.2 Symbols and abbreviations

3.2.1 Abbreviations

Abbreviations used in this standard

Abbreviation	Meaning
B	byte
LBA	logical block address
LSB	Least significant bit
LUN	Logical unit number
MSB	Most significant bit
SBC-4	SCSI Block Commands-4 (see 2.3)
SPC-5	SCSI Primary Commands-5 (see 2.3)

3.2.2 Units

Units used in this standard:

Units	Meaning
μs	microsecond (i.e., 10^{-6} seconds)
m	meter
ms	millisecond (i.e., 10^{-3} seconds)
ns	nanosecond (i.e., 10^{-9} seconds)
s	second (unit of time)

3.2.3 Symbols

Symbols used in this standard:

Symbols	Meaning
Kxx.y	control character
Dxx.y	data character
®	registered trademark

3.2.4 Mathematical operators

Mathematical operators used in this standard:

Mathematical Operators	Meaning
^ or XOR	exclusive logical OR
x	multiplication
/	division
±	plus or minus
≈	approximately
~	approximately equal to
x	multiply
+	add
–	subtract
< or LT	less than
≤ or LE	less than or equal to
= or EQ	equal
≠ or NE	not equal
> or GT	greater than
≥ or GE	greater than or equal to
|v|	the absolute value (i.e., magnitude) of v
Î	set membership

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

3.3.4 may not

keyword that indicates flexibility of choice with no implied preference

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 prohibited

keyword used to describe a feature, function, or coded value that is defined in a a non-SCSI standard (i.e., a standard that is not a member of the SCSI family of standards) to which this standard makes a normative reference where the use of said feature, function, or coded value is not allowed for implementations of this standard

3.3.8 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.

Note 3 to entry: Receipt of reserved code values in defined fields shall be reported as error.

3.3.9 restricted

keyword referring to bits, bytes, words, and fields that are set aside for other identified standardization purposes

Note 1 to entry: A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field in the context where the restricted designation appears.

3.3.10 shall

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

3.3.11 should

keyword indicating flexibility of choice with a strongly preferred alternative

3.3.12 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 the glossary 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). For more information on procedure calls see 3.7.

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 and notes are numbered consecutively throughout this standard.

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 to 9 and/or the upper-case English letters A to 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., 3456FDCA 84BD5E7Ah, 3456FDCA_84BD5E7Ah, 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 to 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

A range of numeric values is represented in this standard in the form “a to z”, where a is the first value included in the range, all values between a and z are included in the range, and z is the last value included in the range (e.g., the representation “0h to 3h” includes the values 0h, 1h, 2h, and 3h).

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;
- c) the thousands separator is used in both the integer portion and the fraction portion of a number; and
- d) the decimal representation for a year is 1999 not 1 999.

Table 1 shows some examples of decimal numbers using various 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 Units of measure

This standard represents values using both decimal units of measure and binary units of measure. Values are represented by the following formats:

- a) for values based on decimal units of measure:
  - 1) numerical value (e.g., 100);
  - 2) space; and
  - 3) prefix symbol and unit:
    - 1) decimal prefix symbol (e.g., M) (see table 2); and
    - 2) unit abbreviation (e.g., B);

  and

- b) for values based on binary units of measure:
  - 1) numerical value (e.g., 1 024);
  - 2) space; and
  - 3) prefix symbol and unit:
    - 1) binary prefix symbol (e.g., Gi) (see table 2); and
    - 2) unit abbreviation (e.g., b).

Table 2 compares the prefix, symbols, and power of the binary and decimal units.

Table 2 — Comparison of decimal prefixes and binary prefixes

Decimal			Binary		
Prefix name	Prefix symbol	Power (base-10)	Prefix name	Prefix symbol	Power (base-2)
kilo	k	10^3	kibi	Ki	2^{10}
mega	M	10^6	mebi	Mi	2^{20}
giga	G	10^9	gibi	Gi	2^{30}
tera	T	10^{12}	tebi	Ti	2^{40}
peta	P	10^{15}	pebi	Pi	2^{50}
exa	E	10^{18}	exbi	Ei	2^{60}
zetta	Z	10^{21}	zebi	Zi	2^{70}
yotta	Y	10^{24}	yobi	Yi	2^{80}

3.6 Bit and byte ordering

In this standard, data structures may be defined by a table. A table defines a complete ordering of elements (i.e., bits, bytes, fields, and dwords) within the structure. The ordering of elements within a table does not in itself constrain the order of storage or transmission of the data structure, but in combination with other normative text in this standard, may constrain the order of storage or transmission of the structure.

In a table, any element that is presented in a row above another element in a lower row is more significant than the lower element, and any element presented to the left of another element in the same row is more significant than the element to the right.

If a table shows bit numbering (see table 3), the least significant bit (LSB) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If a table shows numbering of bytes or characters (see table 4), the most significant byte or character is represented at the lowest number and each less significant byte or character has the next greater number than the immediately more significant byte.

In a field in a table consisting of more than one bit that contains a single value (e.g., a number), the least significant bit (LSB) is shown on the right and the most significant bit (MSB) is shown on the left (e.g., in a byte, bit 7 is the MSB and is shown on the left, bit 0 is the LSB and is shown on the right). The MSB and LSB are not labeled if the field consists of eight or fewer bits. The MSB and LSB are labeled if the field consists of more than eight bits and has no internal structure defined.

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels. Each individual field has an MSB and LSB, but they are not labeled.

In a field containing a text string (e.g., ASCII or UTF-8), only the MSB of the first character and the LSB of the last character are labeled.

Multiple byte fields are represented with only two rows, with the non-sequentially increasing byte number denoting the presence of additional bytes.

A data dword consists of 32 bits. Table 3 shows a data dword containing a single value, where the MSB is on the upper left in bit 31 and the LSB is on the lower right in bit 0.

Table 3 — Example of ordering of bits and bytes within a multi-byte element

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
Note – The Bit x labels in the individual table cells are for reference only and should not appear within tables that use this element format.								

Table 4 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the left and byte 3 (the fourth byte) is on the right. Each byte has an MSB on the left and an LSB on the right.

Table 4 — Example of ordering of bits and bytes within a multiple element

Bit Byte	7	6	5	4	3	2	1	0
0	First byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
1	Second byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
2	Third byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
3	Fourth byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
Note – The Bit x labels in the individual table cells and the xx byte labels in the individual bytes are for reference only and should not appear within tables the use these element formats. In this example the MSB and LSB labels are for reference only, however, they may appear in multi-byte fields as described in this subclause.								

3.7 Notation for procedure calls

In this standard, the model for functional interfaces between entities is a procedure call. 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	A 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 arguments to be specified as inputs and outputs. An interface between entities may require only inputs. If a procedure call has no output arguments, the word OUT, preceding comma, and associated pair of balanced parentheses are omitted.

The following is an example of a procedure call specification:

Found = **Search (**IN (Pattern, Item List), OUT ([Item Found]))

Input arguments:

Pattern: Argument containing the search pattern.

Item List: **Item<NN>** contains the items to be searched for a match.

Output arguments:

Item Found: Item located by the search procedure call. This argument is only returned if the search succeeds.

4 Zoned Block Device Model

4.1 Zoned Block Device model introduction

Table 5 lists topics that apply to zoned block devices.

Table 5 — Zoned block device model topics

Topic [a]	Reference
Zoned block device models	4.2
Zone type models	4.3
Media examples	SBC-4
Logical blocks	SBC-4
Physical blocks	SBC-4
Ready state	SBC-4
Initialization	SBC-4
Format operations	4.5 and SBC-4
Sanitize operations	4.6 and SBC-4
Write protection	SBC-4
Medium defects	SBC-4
Write and unmap failures	SBC-4
Caches	SBC-4
Implicit head of queue command processing	SBC-4
Reservations	4.7, SPC-5, and SBC-4
Error reporting	SBC-4
START STOP UNIT and power conditions	SBC-4
Protection information model	SBC-4
Grouping function	SBC-4
Background scan operations	SBC-4
Association between commands and CbCS permission bits	4.8, SPC-5, and SBC-4
Deferred microcode activation	SBC-4
Model for uninterrupted sequences on LBA ranges	SBC-4
ZBC device internal resource management	4.4.1
Unexpected power removal	4.4.2
Media failure	4.4.3

[a] SBC-4 model topics not listed in this table shall not be supported for host managed zoned block devices. Host aware zoned block devices may support any SBC-4 model topic.

4.2 Zoned Block Device access models

4.2.1 Zoned Block Device access models overview

These are the zoned block device models described in this standard:

a) host aware zoned block device; and
b) host managed zoned block device.

Zoned block devices are accessed using LBAs. The LBAs are divided into ranges called zones (see 4.3). The entire capacity of a zoned block device is organized into a set of contiguous, non-overlapping zones. Figure 1 shows a zoned block device with n zones and m LBAs where LBA 0 is the lowest LBA of zone 0 and LBA m–1 is the highest LBA of zone n–1.

↓ LBA 0 LBA m–1 ↓

zone 0	zone 1	zone 2	…	zone n–1

Figure 1 — Zones in a zoned block device

Zoned block devices use the REPORT ZONES command (see 5.2) for reporting the zone type for each zone.

Table 6 defines the characteristics for the types of zoned block devices defined in this standard.

Table 6 — Characteristics for zoned block devices

Characteristics	Reference	Host aware zoned block device	Host managed zoned block device
Command support	n/a	See SBC-4	See 5.1
PERIPHERAL DEVICE TYPE field value	SPC-5	00h	14h
HAW_ZBC bit value	SBC-4	1b	0b
Conventional zone	4.3.2	Optional	Optional
Sequential write preferred zone	4.3.3.2	Mandatory	Not supported
Sequential write required zone	4.3.3.3	Not supported	Mandatory
REPORT ZONES command	5.2	Mandatory	Mandatory
RESET WRITE POINTER command	5.3	Mandatory	Mandatory

4.2.2 Host aware zoned block device model

A host aware zoned block device reports:

a) the PERIPHERAL DEVICE TYPE field set to 00h (i.e., direct access block device); and
b) the HAW_ZBC bit set to one in the Block Device Characteristics VPD page (see SBC-4).

Host aware zoned block devices:

a) may support conventional zones (see 4.3.2);
b) shall support one or more sequential write preferred zones (see 4.3.3.2); and
c) shall not support sequential write required zones (see 4.3.3.3).

In addition to commands for direct access block devices (see SBC-4), host aware zoned block devices shall support:

a) the REPORT ZONES command (see 5.2); and
b) the RESET WRITE POINTER command (see 5.3).

If a device is a host aware zoned block device, a write command starts in one zone, and the transfer length extends into one or more additional zones, then the device server splits the command into separate write operations for each zone. The write pointer in each write pointer zone (see 4.3.3), if any, is modified only by the write operation within that write pointer zone.

4.2.3 Host managed zoned block device model

A host managed zoned block device reports the PERIPHERAL DEVICE TYPE field set to 14h (i.e., host managed zoned block device).

Host managed zoned block devices:

a) may support conventional zones (see 4.3.2);
b) may support sequential write preferred zones (see 4.3.3.2); and
c) shall support one or more sequential write required zones (see 4.3.3.3).

See 5.1 for a list of commands for host managed zoned block devices.

4.3 Zone type models

4.3.1 Zone type models introduction

In a zone block device, each zone is one of the following types:

a) a conventional zone (see 4.3.2);
b) a sequential write preferred zone (see 4.3.3.2); or
c) a sequential write required zone (see 4.3.3.3).

4.3.2 Conventional zone model

A conventional zone is a type of zone:

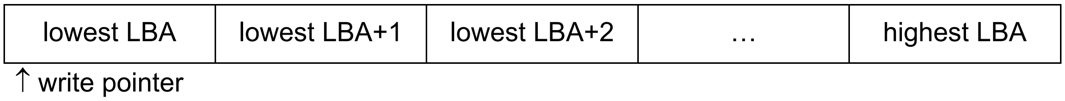
a) that is not associated with a write pointer; and
b) on which the device server does not return the errors described in 4.3.3.3 and performs operations as described in SBC-4.

4.3.3 Write pointer zone models

4.3.3.1 Write pointer zones overview

A write pointer zone is a type of zone that has a write pointer that indicates a location within that zone. Each write pointer zone has one write pointer. That write pointer indicates the LBA that the application client should specify in the LOGICAL BLOCK ADDRESS field of a subsequent write command for that write pointer zone.

Figure 2 shows a write pointer zone where no LBAs have been written since the last reset write pointer operation for that zone. The write pointer is set to the lowest LBA of that zone.



lowest LBA	lowest LBA+1	lowest LBA+2	…	highest LBA

↑ write pointer

Figure 2 — Write pointer zone and write pointer after reset write pointer operation with no subsequent writes

Figure 3 shows a write pointer zone where LBAs have been written since the last reset write pointer operation for that zone (i.e., LBAs less than the write pointer value) and LBAs have not been written since the last reset write pointer operation for that zone (i.e., LBAs greater than or equal to the write pointer value).

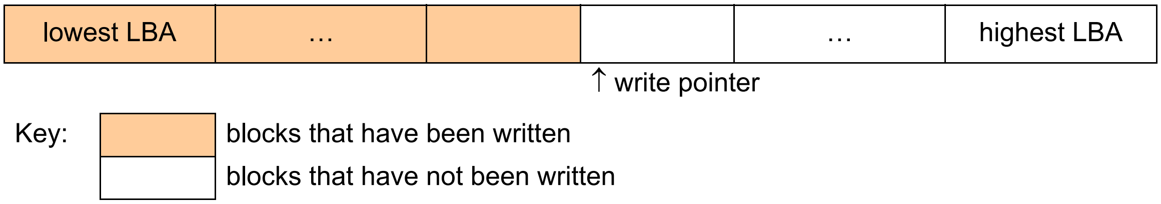

lowest LBA
…
…
highest LBA
↑ write pointer
Key:
blocks that have been written
blocks that have not been written

Figure 3 — Write pointer zone and write pointer

Figure 4 shows the relationship of the write pointer for the following example:

1) at time 1, the write pointer zone starts with the write pointer set to the lowest LBA in that zone;
2) at time 2, a write operation writes data starting at that zone's lowest LBA and sets the write pointer to the last LBA written by the operation plus one;
3) at time 3, a write operation:
    1) writes data that starts at the write pointer position after time 2 and extends to the end of that zone; and
    2) sets the ZONE CONDITION field to FULL (see 5.2.2) for that zone;

    and
4) at time 4, a reset write pointer operation sets the write pointer to the lowest LBA of that write pointer zone.

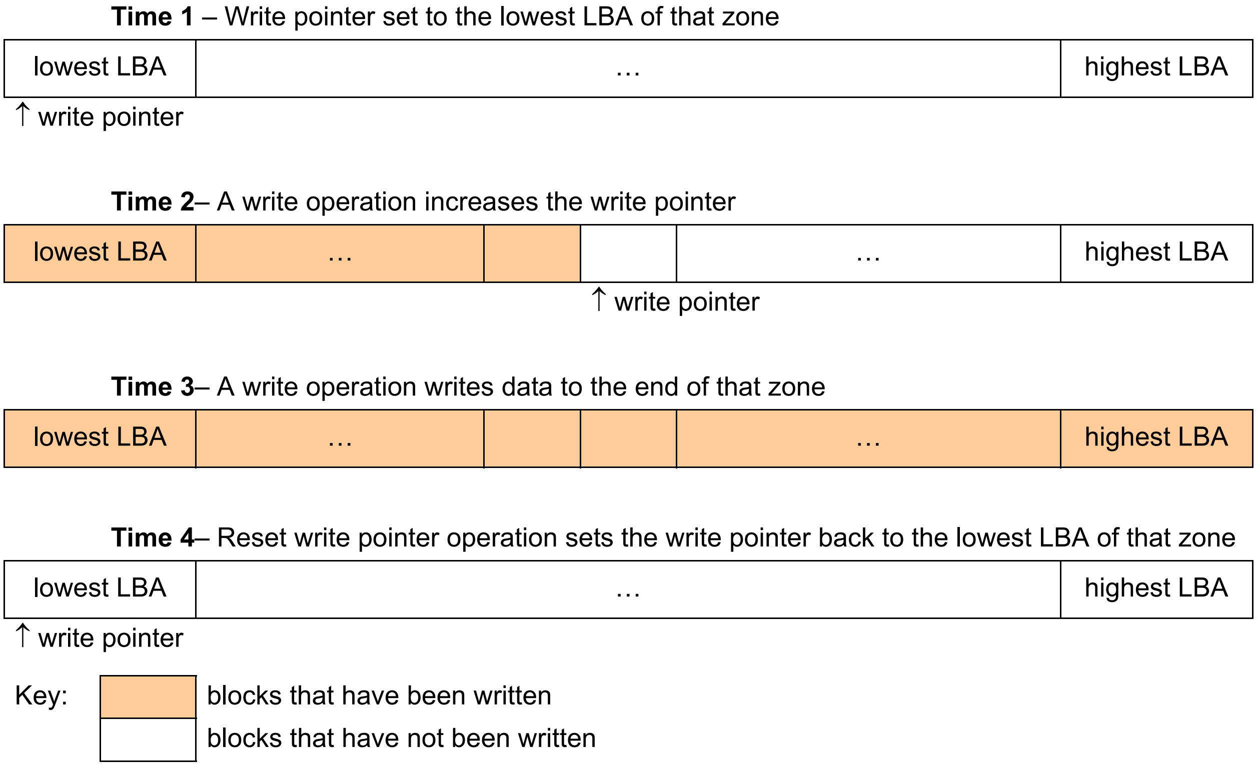

Time 1 – Write pointer set to the lowest LBA of that zone
lowest LBA
…
highest LBA
↑ write pointer
Time 2– A write operation increases the write pointer
lowest LBA
…
…
highest LBA
↑ write pointer
Time 3– A write operation writes data to the end of that zone
lowest LBA
…
…
highest LBA
Time 4– Reset write pointer operation sets the write pointer back to the lowest LBA of that zone
lowest LBA
…
highest LBA
↑ write pointer
Key:
blocks that have been written
blocks that have not been written

Figure 4 — Write pointer zone example operations

A reset write pointer operation sets the write pointer in a write pointer zone to the lowest LBA of that write pointer zone. A reset write pointer operation is requested by:

a) RESET WRITE POINTER command (see 5.3);
b) FORMAT UNIT command (see 4.5 and SBC-4); and
c) SANITIZE command (see 4.6 and SBC-4).

4.3.3.2 Sequential write preferred zone model

A sequential write preferred zone is a write pointer zone (see 4.3.3.1) in which the device server allows write commands that specify a starting LBA that is not equal to the associated zone's write pointer.

Figure 5 shows an example of a write command that starts at the write pointer of a sequential write preferred zone.

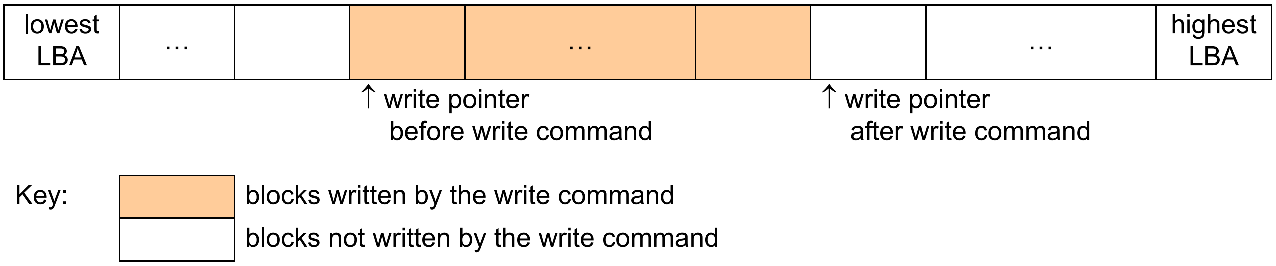
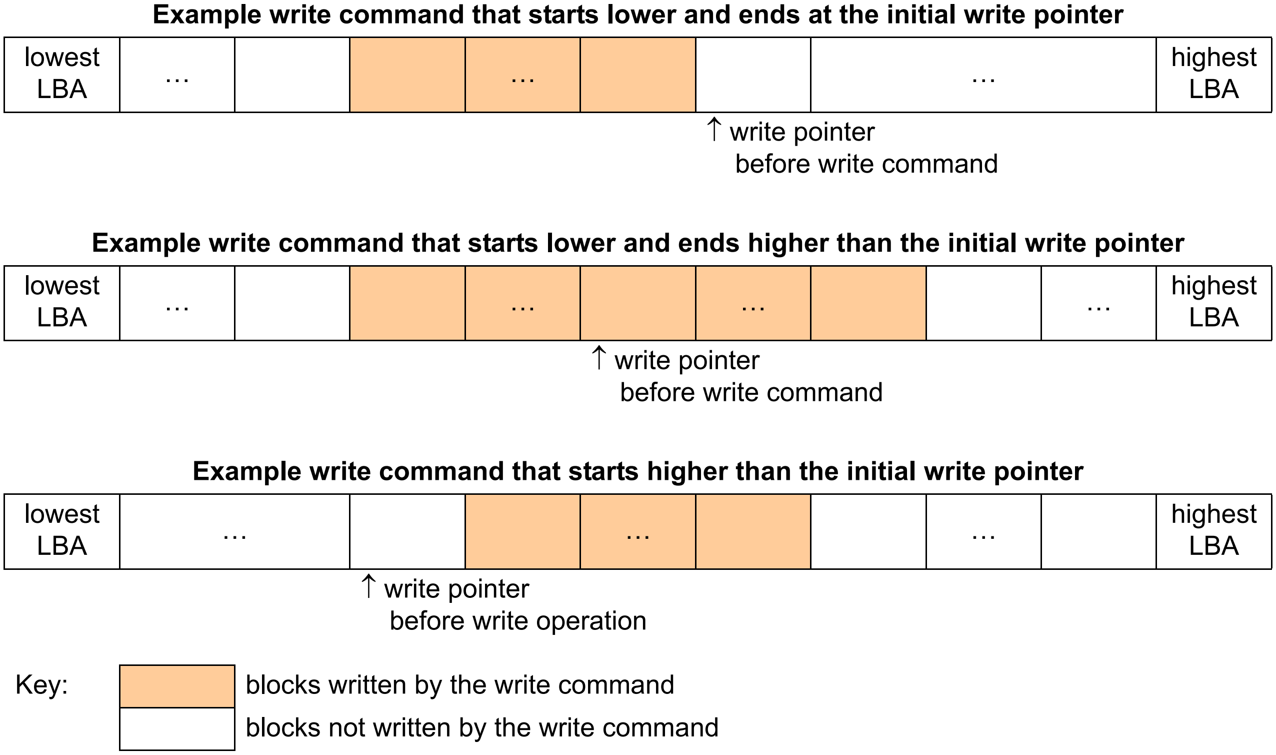

lowest LBA
…
…
…
highest LBA
↑ write pointer before write command
↑ write pointer after write command
Key:
blocks written by the write command
blocks not written by the write command

Figure 5 — Example write command that starts at the write pointer

Figure 6 shows examples of write commands that do not start at the write pointer. The ending value of the write pointer for these examples is outside the scope of this standard.


Example write command that starts lower and ends at the initial write pointer

Example write command that starts lower and ends higher than the initial write pointer

Example write command that starts higher than the initial write pointer

lowest LBA
…
…
…
highest LBA
↑ write pointer before write command
lowest LBA
…
…
…
…
highest LBA
↑ write pointer before write command
lowest LBA
…
…
…
highest LBA
↑ write pointer before write operation
Key:
blocks written by the write command
blocks not written by the write command

Figure 6 — Examples of write commands that do not start at the write pointer

If a write operation encounters a write error that has not written all of the LBAs (e.g., a write command terminates with CHECK CONDITION status with sense key set to MEDIUM ERROR or HARDWARE ERROR), then the write pointer is set to a vendor specific write pointer value.

If write caching is enabled (i.e., WCE bit is set to one, see SBC-4), a write command completes without error, and a deferred write error occurs for a write operation associated with that command, then the write pointer is set to a vendor specific write pointer value.

A read operation on an LBA that has been written since the last reset write pointer operation in that sequential write preferred zone shall return the last written data. A read operation on an LBA that has not been written since the last reset write pointer operation in that sequential write preferred zone shall return user data set to the last initialization pattern that was set by:

a) the manufacturer's default initialization pattern;
b) the initialization pattern set by the last FORMAT UNIT command (see SBC-4), if any; or
c) the initialization pattern set by the last SANITIZE command with the service action set to OVERWRITE (see SBC-4), if any.

A verify operation on an LBA that has been written since the last reset write pointer operation in that sequential write preferred zone shall use the last written data for that verify operation. A verify operation on an LBA that has not been written since the last reset write pointer operation in that sequential write preferred zone shall use user data set to the last initialization pattern that was set by:

a) the manufacturer's default initialization pattern;
b) the initialization pattern set by the last FORMAT UNIT command (see SBC-4), if any; or
c) the initialization pattern set by the last SANITIZE command with the service action set to OVERWRITE (see SBC-4), if any.

4.3.3.3 Sequential write required zone model

A sequential write required zone is a write pointer zone (see 4.3.3.1) in which the device server requires that write commands specify a starting LBA that is equal to that zone’s write pointer.

If the device server processes a write command without error, then the write pointer is modified to indicate the LBA where a subsequent write operation within the zone shall be performed.

If the ending LBA of a write command that completed without error is equal to the highest LBA of a host managed zone, then the ZONE CONDITION field is set to FULL (see 5.2.2) and the write pointer is invalid.

Editor’s Note 1: Doesn't the above rule apply to write preferred zones too? If yes, shouldn't this text be moved to see 4.3.3.1?

A reset write pointer operation (see 4.3.3.1) is the only method for causing an invalid write pointer to become valid.

If the device server processes a write command that is terminated with CHECK CONDITION status, then the value of the write pointer may be unknown to the application client.

Editor’s Note 2: Some places say 'may be unknown' other say 'vendor specific', which is right?

If the device server processes a read command with:

a) the starting LBA in a sequential write required zone; and
b) the ending LBA is in the same sequential write required zone and is less than the write pointer,

then the device server shall perform read operations for that command.

If the device server processes a write command with:

a) the starting LBA equal to the write pointer;
b) the ending LBA is in the same sequential write required zone; and
c) an ending LBA that is equal to the last logical block within a physical block,

then the device server shall perform write operations for that command.

If the device server processes a write command with:

a) the starting LBA in a sequential write required zone that is not equal to the write pointer; or
b) an ending LBA that is not equal to the last logical block within a physical block (see SBC-4),

then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to UNALIGNED WRITE COMMAND.

Editor's Note 3: Shouldn't the device server be required to not write any data in the above case?

If the device server processes a write command with an ending LBA that is not in the same sequential write required zone as the starting LBA, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to WRITE BOUNDARY VIOLATION.

Editor's Note 4: Shouldn't the device server be required to not write any data in the above case?

If the device server processes a read command with the ending LBA greater than or equal to the write pointer, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to ATTEMPT TO READ INVALID DATA.

If the device server processes a read command with:

a) a starting LBA that does not cause the command to be terminated (e.g., the starting LBA is less than the write pointer or the zone is full); and
b) the ending LBA is not in the same sequential write required zone as the starting LBA,

then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to READ BOUNDARY VIOLATION,

If the device server terminates a command with the additional sense code set to:

a) UNALIGNED WRITE COMMAND;
b) WRITE BOUNDARY VIOLATION; or
c) ATTEMPT TO READ INVALID DATA,

then the device server shall return the write pointer associated with the sequential write required zone specified by the LOGICAL BLOCK ADDRESS field of that command in the INFORMATION field of the sense data (see SPC-5).

If the device server processes a write command with the starting LBA that is not in a sequential write required zone and specifies an LBA that is in a sequential write required zone, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to WRITE BOUNDARY VIOLATION.

Editor's Note 5: Is any data written in the above case?

If the device server processes a read command with the starting LBA that is not in a sequential write required zone and specifies an LBA that is in a sequential write required zone, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to ATTEMPT TO READ INVALID DATA.

4.4 Zoned block device extensions to block device model

4.4.1 Zoned block device internal resource management

A zoned block device requires internal resources to maintain each zone. Insufficient resources may result in degraded functionality (e.g., reduced performance, increased power consumption, or increased reporting of write errors).

The device server uses the RESET bit in each zone descriptor (see 5.2.2) to report that an insufficient resources condition has been detected for the indicated zone.

The application client clears the insufficient resources condition by sending a RESET WRITE POINTER command that specifies the affected zone to the device server. Before sending the RESET WRITE POINTER command, the application client may copy the logical block data in the affected zone to another location.

4.4.2 Unexpected power removal

If power is removed from a zoned block device prior to the device server completing the processing of a power loss expected event (see SAM-5) and:

a) there are partially completed write operations;
b) there is logical block data in volatile write cache; or
c) write pointers for completed write operations are not saved to a nonvolatile location,

then an unexpected power removal condition has occurred.

An unexpected power removal condition may cause write pointers for partially completed write commands to be updated to reflect the partial logical block data written to the medium (e.g., the LBA plus one of the last logical block that the device server attempted to write).

If after an unexpected power removal condition the application client requests the parameter data returned by a REPORT ZONES command (see 5.2), then the write pointers in the parameter data may not correspond to the ending LBA plus one of the last command to report completion in the write pointer zone.

4.4.3 Media failure

Editor's Note 6: Is media failure handling restricted to host managed zoned block devices?

Failures may cause some or all of the host managed zoned block device media to become inaccessible. If host managed zoned block device media is still available to be used after a failure, then the media may continue to work properly.

If a host managed zoned block device has a media failure, then the device server may indicate that zones associated with the media failure are unavailable for reading or writing by setting the ZONE CONDITION field in the REPORT ZONES parameter data (see 5.2.2) to OFFLINE (see table 15).

If a ZONE CONDITION field in the REPORT ZONES parameter data is set to OFFLINE, then the device server shall terminate commands that attempt to read or write in the associated zone with CHECK CONDITION status, with the sense key set to MEDIUM ERROR and the additional sense code set to ZONE IS OFFLINE.

4.4.4 Host managed zoned block device extensions to block device caches model

Editor's Note 7: In a separate proposal

4.5 Format operations

All of the functions defined for the FORMAT UNIT command (see SBC-4) are available for zoned block devices. A FORMAT UNIT command affects all zones, however some characteristics are dependent on the zone type (see 4.3) present in the zoned block device. A zoned block device may support multiple types of zones.

For each conventional zone (see 4.3.2), a format operation is performed as specified in SBC-4.

For each write pointer zone (see 4.3.3), a format operation is performed as specified in SBC-4 and a reset write pointer operation (see 4.3.3.1) is performed

4.6 Sanitize operations

All of the functions defined for the SANITIZE command (see SBC-4) are available for zoned block devices. A SANITIZE command affects all zones, however some characteristics are dependent on the zone type (see 4.3) present in the zoned block device. A zoned block device may support multiple types of zones.

For each conventional zone (see 4.3.2), a sanitize operation is performed as specified in SBC-4.

For each write pointer zone (see 4.3.3), a sanitize operation is performed as specified in SBC-4 with the following additional requirements:

a) a successful sanitize operation shall include performing a reset write pointer operation (see 4.3.3.1); and
b) a failed sanitize operation followed by a SANITIZE command with the service action of EXIT FAILURE MODE should result in the reset write pointer operations being successful.

4.7 Reservations

Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation. See SPC-5 for a description of reservations for commands described in that standard. See SBC-4 for a description of reservations for commands described in that standard. See table 7 for a description of reservations for commands described in this standard.

Commands from I_T nexuses holding a reservation should complete normally. Table 7 specifies the behavior of commands from registered I_T nexuses when a registrants only or all registrants type persistent reservation is present.

For each command, this standard or SPC-5 defines the conditions that result in the device server completing the command with RESERVATION CONFLICT status.

Table 7 — ZBC commands that are allowed in the presence of various reservations

Command	Addressed logical unit has this type of persistent reservation held by another I_T nexus				
	From any I_T nexus		From registered I_T nexus (RR all types)	From I_T nexus not registered	
	Write Exclusive	Exclusive Access		Write Exclusive - RR	Exclusive Access - RR
REPORT ZONES	Allowed	Conflict	Allowed	Allowed	Conflict
RESET WRITE POINTER	Conflict	Conflict	Allowed	Conflict	Conflict

Key:
RR = Registrants Only or All Registrants
Allowed = Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present should complete normally.
Conflict = Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present shall not be performed, and the device server shall complete the command with RESERVATION CONFLICT status.

4.8 Association between commands and CbCS permission bits

Table 8 defines the CbCS permissions required for each command defined in this standard. The permissions shown in table 8 are defined in the PERMISSIONS BIT MASK field in the CbCS capability descriptor in a CbCS extension descriptor (see SPC-5). This standard does not define any permissions specific to zoned block commands.

Table 8 — Associations between commands and CbCS permissions

Command	Permissions bit mask bits [a]				
	DATA READ	DATA WRITE	PARM READ	PARM WRITE	PHY ACC
REPORT ZONES			1		
RESET WRITE POINTER		1			

[a] A device server shall only process a command shown in this table as specified by the CDB field of an extended CDB (see SPC-5) that contains a CbCS capability descriptor when all of the bits marked with a “1” in the row for that command are set to one in the PERMISSIONS BIT MASK field in that descriptor. The permissions bits represented by the empty cells in a row are ignored. If a device server receives a command specified by the CDB field of an extended CDB that does not contain the CbCS capability descriptor with all of the bits set to one as defined in this table, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

5 Commands for zoned block devices

5.1 Commands for zoned block devices overview

The commands for host aware zoned block devices are defined in SBC-4.

The commands for host managed zoned block devices are listed in table 9.

Table 9 — Commands for host managed zoned block devices (part 1 of 2)

Command	Type	LBACT	Reference
ATA PASS-THROUGH (12)	O	n/a	SAT-3
ATA PASS-THROUGH (16)	O	n/a	SAT-3
FORMAT UNIT	O		SBC-3
INQUIRY	M	n/a	SPC-4
LOG SELECT	O		SPC-4
LOG SENSE	M		SPC-4
MODE SELECT (10)	M	n/a	SPC-4
MODE SENSE (10)	M	n/a	SPC-4
PERSISTENT RESERVE IN	O		SPC-4
PERSISTENT RESERVE OUT	O		SPC-4
READ (16)	M	R	SBC-3
READ BUFFER	O		SPC-4
READ CAPACITY (16)	M	n/a	SBC-3
READ DEFECT DATA (12)	O		SBC-3
REPORT LUNS	M	n/a	SPC-4
REPORT SUPPORTED OPCODES	M	n/a	SPC-4
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	M	n/a	SPC-4
REPORT TIMESTAMP	O		SPC-4
REPORT ZONES	M	n/a	5.2
REQUEST SENSE	M	n/a	SPC-4
RESET WRITE POINTER	M	n/a	5.3
SET TIMESTAMP	O		SPC-4

Key:
O = optional
M = mandatory
X = implementation requirements are defined in the reference
R = read command
U = unmap command
V = verify command
W = write command
Z = other command
PI = protection information
LBACT = logical block access command type (see SBC-4)

Table 9 — Commands for host managed zoned block devices (part 2 of 2)

Command	Type	LBACT	Reference
SANITIZE	O	Z	SBC-3
SECURITY PROTOCOL IN	O	n/a	SPC-4
SECURITY PROTOCOL OUT	O	n/a	SPC-4
SEND DIAGNOSTIC	O	n/a	SPC-4
START STOP UNIT	M	n/a	SBC-3
SYNCHRONIZE CACHE (16)	M	W	SBC-3
TEST UNIT READY	M	n/a	SPC-4
VERIFY (16)	O	n/a	SBC-3
WRITE (16)	M	W	SBC-3
WRITE BUFFER	O		SPC-4
WRITE SAME (16)	M	U, W	SBC-3

Key:

O =	optional	V =	verify command
M =	mandatory	W =	write command
X =	implementation requirements are defined in the reference	Z =	other command
		PI =	protection information
R =	read command	LBACT =	logical block access command type (see SBC-4)
U =	unmap command		

5.2 REPORT ZONES command

5.2.1 REPORT ZONES command introduction

The REPORT ZONES command (see table 10) requests that the device server return the zone structure of the zoned block device.

Table 10 — REPORT ZONES command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Eh)							
1	Reserved			SERVICE ACTION (14h)				
3	(MSB)	ZONE START LBA						
•••								
9								(LSB)
10	(MSB)	ALLOCATION LENGTH						
•••								
13								(LSB)
14	Reserved				REPORTING OPTIONS			
15	CONTROL							

The OPERATION CODE field and SERVICE ACTION field are defined in SPC-4 and shall be set to the values shown in table 10 for the REPORT ZONES command.

The ZONE START LBA field specifies the starting LBA of the first zone to be reported. If the ZONE START LBA field does not specify the starting LBA of a zone, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The ALLOCATION LENGTH field is defined in SPC-4.

The REPORTING OPTIONS field (see table 11) specifies the information to be returned in the parameter data.

Table 11 — REPORT ZONES REPORTING OPTIONS field

Code	Description
0h	List the zones in the zoned block device for all values in the ZONE CONDITION field and RESET bit.
1h	List the full zones (i.e., the zone condition for the zone is FULL (see table 15)).
2h	List the open zones (i.e., the zone condition for the zone is OPEN (see table 15)).
3h	List the empty zones (i.e., the zone condition for the zone is EMPTY (see table 15)).
4h	List the read only zones (i.e., the zone condition for the zone is READ ONLY (see table 15)).
5h	List the offline zones (i.e., the zone condition for the zone is OFFLINE (see table 15)).
6h	List the zones for which the RESET bit is set to one (see table 13).
7h to Fh	Reserved

Editor’s Note 8: Should there be a code defined to list only the conventional zones that are not READ ONLY and not OFFLINE.

The CONTROL byte is defined in SAM-5.

5.2.2 REPORT ZONES parameter data

The REPORT ZONES parameter data is defined in table 12.

Table 12 — REPORT ZONES parameter data

Bit Byte	7	6	5	4	3	2	1	0
1	(MSB)	ZONE LIST LENGTH (n–63)						
•••								
3								(LSB)
4	Reserved							SAME
5	Reserved							
•••								
63								
Zone descriptors list								
64	Zone descriptor [first]							
•••								
127								
	⋮							
n–63	Zone descriptor [last]							
•••								
n								

The ZONE LIST LENGTH field contains the length in bytes of the zone descriptors list. The zone list length is the number of zones reported multiplied by 64. The content of the ZONE LIST LENGTH field is not altered based on the allocation length (see SPC-4).

A SAME bit set to zero indicates that the zone type and zone length in each zone descriptor may be different. A SAME bit set to one indicates that the zone type and zone length in each zone descriptor are equal to the zone type and zone length indicated in the first zone descriptor.

Each zone descriptor (see table 13) contains the description for one zone. The zone descriptors shall be sorted in ascending order based on the zone start LBA value.

Table 13 — Zone descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			RESET
2	Reserved							
•••								
7								
8	(MSB)	ZONE LENGTH						
•••								
15								(LSB)
16	(MSB)	ZONE START LBA						
•••								
23								(LSB)
24	(MSB)	WRITE POINTER LBA						
•••								
31								(LSB)
32	Reserved							
•••								
63								

The ZONE TYPE field indicates the type of zone as shown in table 14.

Table 14 — Zone descriptor ZONE TYPE field

Code	Description
0h	Reserved
1h	Conventional (see 4.3.2)
2h	Sequential write required (see 4.3.3.3)
3h	Sequential write preferred (see 4.3.3.2)
4h to Fh	Reserved

The ZONE CONDITION field indicates the condition of the zone as described in table 15.

Table 15 — Zone descriptor ZONE CONDITION field

Code	Name	Applies to zone type (see table 14)	Description
0h			Reserved
1h	EMPTY	sequential write preferred and sequential write required	The device server has not performed a write operation to this writer pointer zone since the write pointer was set to the lowest LBA of this zone. This zone is available to perform read operations and write operations.
2h	OPEN	sequential write preferred and sequential write required	The device server has attempted a write operation to this writer pointer zone since the write pointer was set to the lowest LBA of this zone and the zone condition is not FULL. This zone is available to perform read operations and write operations.
3h to Ch			Reserved
Dh	READ ONLY	all	Only read operations are allowed in this zone. The WRITE POINTER LBA field is invalid. The device server shall terminate any command that attempts a write operation in this zone with CHECK CONDITION status with the sense key set to DATA PROTECT and additional sense code set to ZONE IS READ ONLY.
Eh	FULL	sequential write preferred and sequential write required	All logical blocks in this writer pointer zone contain logical block data. The WRITE POINTER LBA field is invalid.
Fh	OFFLINE	all	Read commands and write commands shall be terminated as described in 4.4.3. The WRITE POINTER LBA field is invalid.

Editor’s Note 9: There are not codes defined in this table that are clearly the correct choice for a conventional zone. Maybe the right answer is to define a specific value to be returned for a conventional zone that is not offline.

A RESET bit set to zero indicates that the device server has sufficient resources to maintain the zone described by this zone descriptor. A RESET bit set to one indicates that the application client should send a RESET WRITE POINTER command (see 5.3) that specifies the zone described by this zone descriptor as described in 4.4.1. If the zone type is conventional or sequential write preferred (see table 14), then the device server shall not set the RESET bit to one. The device server shall set the RESET bit to zero for conventional zones.

The ZONE LENGTH field indicates the number of logical blocks in this zone.

The ZONE START LBA field indicates the lowest numbered LBA in this zone.

The WRITE POINTER LBA field indicates the starting LBA that the application client should specify in the next write command associated with this zone. The content of the WRITE POINTER LBA field is not valid if:

a) the zone type is conventional (see table 14); or
b) the content of the ZONE CONDITION field (see table 15) indicate that the WRITE POINTER LBA field is invalid.

If the zone type is set to 1h (i.e., conventional), then the WRITE POINTER LBA field is ignored by the application client.

5.3 RESET WRITE POINTER command

The RESET WRITE POINTER command (see table 16) performs one or more reset write pointer operations (see 4.3.3.1).

Table 16 — RESET WRITE POINTER command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Fh)							
1	Reserved			SERVICE ACTION (14h)				
3	(MSB)							
•••	ZONE ID							
9								(LSB)
10								
•••	Reserved							
13								
14	Reserved							RESET ALL
15	CONTROL							

The OPERATION CODE field and SERVICE ACTION field are defined in SPC-4 and shall be set to the values shown in table 16 for the RESET WRITE POINTER command.

The ZONE ID field specifies the zone start LBA, if any, of the write pointer zone on which the device server shall perform a reset write pointer operation (see 4.3.3.1).

If the RESET ALL bit is set to zero and:

a) the ZONE ID field does not specify the zone start LBA of a write pointer zone (see 4.3.3), then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB;
b) the ZONE CONDITION is set to READ ONLY (see table 15) in the REPORT ZONES parameter data (see 5.2.2) for the zone specified by the ZONE ID field, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT and additional sense code set to ZONE IS READ ONLY; and
c) the ZONE CONDITION is set to OFFLINE (see table 15) in the REPORT ZONES parameter data (see 5.2.2) for the zone specified by the ZONE ID field, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT and additional sense code set to ZONE IS OFFLINE.

A RESET ALL bit set to one specifies that the device server shall perform a reset write pointer operation (see 4.3.3.1) on all OPEN zones and FULL zones (see table 15). If the RESET ALL bit is set to one, then the device server shall ignore the ZONE ID field.

The CONTROL byte is defined in SAM-5.

6 Parameters for zoned block devices

6.1 Diagnostic parameters

The diagnostic pages and their corresponding page codes for host aware zoned block devices are defined in SPC-5 and SBC-4.

The diagnostic pages and their corresponding page codes for host managed zoned block devices are defined in table 17.

Table 17 — Diagnostic page codes for host managed zoned block devices

Diagnostic page name	Page code	Reference
Diagnostic pages assigned by SPC-5	30h to 3Fh	SPC-5
Direct access device diagnostic pages	40h to 7Fh	SBC-4
SCSI enclosure services diagnostic pages	01h to 2Fh	SES-4
Vendor specific diagnostic pages	80h to FFh	

6.2 Log parameters

The log pages and their corresponding page codes and subpage codes for host aware zoned block devices are defined in SPC-5 and SBC-4.

The log pages and their corresponding page codes and subpage codes for host managed zoned block devices are defined in table 18. See SPC-5 for a detailed description of logging operations.

Table 18 — Log page codes and subpage codes for host managed zoned block devices

Log page name	Page code [a]	Subpage code [a]	Reference
Application Client	0Fh	00h	SPC-5
Background Scan Results	15h	00h	SBC-4
Informational Exceptions	2Fh	00h	SPC-5
Power Condition Transitions	1Ah	00h	SPC-5
Protocol-Specific Port	18h	00h to FEh	SPC-5
Read Error Counters	03h	00h	SPC-5
Self-Test Results	10h	00h	SPC-5
Start-Stop Cycle Counter	0Eh	00h	SPC-5
Supported Log Pages	00h	00h	SPC-5
Supported Log Pages and Subpages	00h	FFh	SPC-5
Supported Subpages	01h to 3Fh	FFh	SPC-5
Temperature	0Dh	00h	SPC-5
Verify Error Counters	05h	00h	SPC-5
Write Error Counters	02h	00h	SPC-5
Vendor specific	30h to 3Eh	00h to FEh	n/a

[a] All page code and subpage code combinations not shown in this table are reserved for host managed zoned block devices.

6.3 Mode parameters

The mode pages and their corresponding page codes and subpage codes for host aware zoned block devices are defined in SPC-5 and SBC-4.

The mode pages and their corresponding page codes and subpage codes for host managed zoned block devices are defined in table 19. See SBC-4 for mode parameter block descriptors used by host managed zoned block devices.

Table 19 — Mode page codes and subpage codes for host managed zoned block devices

Mode page name	Page code	Subpage code	Reference
Application Tag	0Ah	02h	SBC-4
Background Control	1Ch	01h	SBC-4
Caching	08h	00h	SBC-4
Control	0Ah	00h	SPC-5
Control Extension	0Ah	01h	SPC-5
Disconnect-Reconnect	02h	00h	SPC-5
Informational Exceptions Control	1Ch	00h	SBC-4
Power Condition	1Ah	00h	SPC-5
Protocol-Specific Logical Unit	18h	00h	SPC-5
Protocol-Specific Port	19h	00h	SPC-5
Read-Write Error Recovery	01h	00h	SBC-4
Return all mode pages and subpages [a]	3Fh	FFh	SPC-5
Return all mode pages only (i.e., not including subpages) [a]	3Fh	00h	SPC-5
Return all subpages for the specified mode page code [a]	00h to 3Eh	FFh	SPC-5
Verify Error Recovery	07h	00h	SBC-4
Vendor specific [b]			
Reserved	all other combinations		

[a] Valid only for the MODE SENSE command.

[b] These vendor specific mode page code and subpage code combinations do not require a page format: mode page 00h with subpage code 00h and mode page codes 20h to 3Eh with all subpage codes.

6.4 Vital product data (VPD) parameters

The VPD pages and their corresponding page codes for host aware zoned block devices are defined in SPC-5 and SBC-4.

Editor's Note 10: The Block Device Characteristics VPD page is optional in SBC-4, but the HAW_ZBC bit is in that VPD page and the HAW_ZBC bit is required to identify a host aware zoned block devices.

The VPD pages and their corresponding page codes for host managed zoned block devices are defined in table 20 and in SPC-5.

Table 20 — VPD page codes for host managed zoned block devices

VPD page name	Page code [a]	Reference	Support requirements
ATA Information	89h	SAT-3	See SAT-3
Block Device Characteristics	B1h	SBC-4	See SBC-4
Block Limits	B0h	SBC-4	See SBC-4

[a] All page codes not shown in this table or SPC-5 are reserved for host managed zoned block devices.

Annex A

(Informative)

Example zoned block devices

A.1 TBD

Bibliography

Editor’s Note 11: These are example bibliography entries. Real ones need to be inserted when the time comes

ISO/IEC 14776-153, *Serial Attached SCSI - 2.1 (SAS-2.1).*

ISO/IEC 14776-261, *SAS Protocol Layer (SPL).*

ISO/IEC 14776-323, *SCSI Block Commands - 3 (SBC-3).*

ISO/IEC 14776-333, *SCSI Streaming Commands - 3 (SSC-3).*

ISO/IEC 14776-365, *Multi-Media Commands - 5 (MMC-5).*