International Standard

ISO/IEC 14776-346

First edition
2024-09

Information technology — Small computer system interface (SCSI) —

Part 346: Zoned Block Commands - 2 (ZBC-2)

Technologies de l'information — Interface de petit système d'ordinateur (SCSI) —

Partie 346: Commandes de blocs de zones - 2 (ZBC-2)

Reference number
ISO/IEC 14776-346:2024(en)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Foreword

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, ISO and IEC had not received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by INCITS (as INCITS 550-2023) and drafted in accordance with its editorial rules. It was assigned to Joint Technical Committee ISO/IEC JTC 1, *Information technology*, and adopted under the “fast-track procedure”.

A list of all parts in the ISO/IEC 14776 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Contents

Tables

Figures

Foreword (This foreword is not part of American National Standard INCITS 550-2023.)

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 – 6 (SPC-6) and the SCSI Block Commands standard – 5 (SBC-5) to facilitate operation of 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, 700 K Street NW, Suite 600, Washington, DC 20001.

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:

Laura Lindsay, Chair
Anish Karmarkar, Vice-Chair
Jennifer Garner, Secretary

Organization Represented	*Name of Representative*
Adobe Systems, Inc.	Scott Foshee
	David Sankel (Alt.)
AIM Global, Inc.	Steve Halliday
	Mary Lou Bosco (Alt.)
	Chuck Evanhoe (Alt.)
Amazon Web Services, Inc.	Tyler Messa
	Roland Blanding (Alt.)
	Rich Gardner (Alt.)
	Judd Hesselroth (Alt.)
Apple	Helene Workman
	Hicham Lozi (Alt.)
	Anna Weinberg (Alt.)
Dell, Inc.	Erin Bournival
Department of Commerce - NIST	Michael Hogan
	Lisa Carnahan (Alt.)
	Jessica Dickson (Alt.)
	Sal Francomacaro (Alt.)
	Suzanne Radell (Alt.)
Edison Design Group	John Spicer
	Daveed Vandevoorde (Alt.)
Farance, Inc.	Frank Farance
	Timothy Schoechle (Alt.)
Federal Bureau of Investigation (FBI)	Robert Brown
	Tara Alhariri (Alt.)
	Darrin Jones (Alt.)
	Kathleen Noyes (Alt.)
Futurewei Technologies, Inc.	Wael Diab
	Wilbert Adams (Alt.)
	Timothy Jeffries (Alt.)
GDCE, Inc.	Scott Payton
George Mason University	Duminda Wijesekera
Google	Catherine Nelson
	Ryan Wagner (Alt.)
	Lenora Zimmerman (Alt.)
GS1GO	Steven Keddie
	Edward Merrill (Alt.)
	Dan Mullen (Alt.)
HP, Inc.	Muhammad Ali
	Paul Jeran (Alt.)
IBM Corporation	Arnaud Le Hors
	Alexander Tarpinian (Alt.)

Organization Represented	*Name of Representative*
Immersion4 LLC	Serge Conesa
	Dawn Blackbird (Alt.)
Intel Corporation	Philip Wennblom
	Grace Wei (Alt.)
Microsoft Corporation	Laura Lindsay
	John Calhoon (Alt.)
NVidia Corporation	Timothy Costa
	Bryce Adelstein Lelbach (Alt.)
Oracle	Anish Karmarkar
	Michael Kavanaugh (Alt.)
	Luke Kowalski (Alt.)
	Jan-Eike Michels (Alt.)
	Elaine Newton (Alt.)
Qualcomm, Inc.	Michael Atlass
	Mazen Chmaytelli (Alt.)
	Osok Song (Alt.)
Sandflow Consulting LLC	Pierre-Anthony Lemieux
United States Dept. of Defense	Vernita Harris
	Matthew Ivey (Alt.)
	Tejas Patel (Alt.)

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

William Martin, Chair
Curtis Ballard, Vice-Chair
Curtis Stevens, Secretary
Frederick Knight, International Representative

Organization Represented	*Name of Representative*
Amphenol Corporation	Paul Coddington
	Brad Brubaker (Alt.)
	Zhineng Fan (Alt.)
	Adrian Green (Alt.)
	Donald Harper (Alt.)
	Michael Klempa (Alt.)
	Sam Kocsis (Alt.)
	Martin Li (Alt.)
	Chris Lyon (Alt.)
	Alex Persaud (Alt.)
	Michael Scholeno (Alt.)
	Hu Silver (Alt.)
	Michael Wingard (Alt.)
	Matt Wright (Alt.)
Anritsu Corporation	Tadanori Nishikobara
	Yuji Sekine (Alt.)
Broadcom, Inc.	Brad Besmer
	Srikiran Dravida (Alt.)
	John Gardner (Alt.)
	Jeffrey Gauvin (Alt.)
	Rick Kutcipal (Alt.)
	Bernhard Laschinsky (Alt,)
	David Peterson (Alt.)
	James Smart (Alt.)
	Jason Stuhlsatz (Alt.)
	Steven Wilson (Alt.)
Dell, Inc.	Kevin Marks
	David Black (Alt.)
	George Ericson (Alt.)
	Christopher Goonan (Alt.)
	Bill Lynn (Alt.)
	Gary Kotzur (Alt.)
	Bill Lynn (Alt.)
	Marlon Ramroopsingh (Alt.)

Organization Represented	*Name of Representative*
Flextronics International (Taiwan), Ltd.	Roger Wan
	Istvan BakroNagy (Alt.)
Foxconn Interconnect Technology, Ltd. (FIT)	Fred Fons
	Gary Hsieh (Alt.)
	Glenn Moore (Alt.)
	Miller Zhao (Alt.)
Futurewei Technologies, Inc.	Victor Cao
	Timothy Jeffries (Alt.)
	Nelson Liao (Alt.)
Google	Radha Ramachandran
	Thieu Le (Alt.)
	Chris Sabol (Alt.)
	Bart Van Assche (Alt.)
Hewlett Packard Enterprise	Curtis Ballard
	Chris Cheng (Alt.)
	Robert Elliott (Alt.)
	Jeff Wolford (Alt.)
IBM Corporation	Kevin Butt
	Mike Osborne (Alt.)
Intel Corporation	Michael Hoard
	Phil Cayton (Alt.)
	Kim Malone (Alt.)
	Minesh Patel (Alt.)
	Eric Pickering (Alt.)
	Tzewen Wang (Alt.)
	Juntao Yuan (Alt.)
Keysight Technologies, Inc.	Vincent Yew
Kioxia Corporation	John Geldman
	Cameron Brett (Alt.)
	Mark Carlson (Alt.)
	Don Harwood (Alt.)
	Koichi Nagai (Alt.)
	Paul Suhler (Alt.)
	Tatsuya Tanaka (Alt.)
LUXSHARE-ICT, Inc.	Scott Shuey
	Josue Castillo (Alt.)
	Jinhua Chen (Alt.)
	Pat Young (Alt.)
Marvell Semiconductor, Inc.	Craig Carlson
	Wei Liu (Alt.)
	James Walch (Alt.)
Meta Platforms, Inc.	Ross Stenfort
Microchip Technology, Inc.	Jermiah Tussey
	Sapna Devi (Alt.)
	Sanjay Goyal (Alt.)
	Adnan Jiwani (Alt.)
	Chethen K (Alt.)
	Keith Shaw (Alt.)
	Ariel Sibley (Alt.)
	Tim Symons (Alt.)
	Gregory Tabor (Alt.)
Molex, Inc.	Alexandra Haser
	Darian Schulz (Alt.)
	Scott Sommers (Alt.)
NetApp, Inc.	Frederick Knight
	Chris Fore (Alt.)
	Jaimon George (Alt.)
	John Meneghini (Alt.)
Oracle	Martin Petersen
	Jon Allen (Alt.)
	Ioannis Psyllas (Alt.)
QNAP Systems, Inc.	Ming-chih Chang
	CH Yang (Alt.)
Quantum Corporation	Darryl Torske
	Carsten Prigge (Alt.)

Organization Represented	*Name of Representative*
Samsung Semiconductor, Inc. (SSI)	William Martin
	Judy Brock (Alt.)
	HeeChang Cho (Alt.)
	MiKyeong Kang (Alt.)
	Sung Lee (Alt.)
Seagate Technology	Gerald Houlder
	Umamahesh Allenki (Alt.)
	James Borden (Alt.)
	Andre Boyogueno (Alt.)
	Alvin Cox (Alt.)
	Ian Davies (Alt.)
	John Fleming (Alt.)
	Jim Hatfield (Alt.)
	Gerald Houlder (Alt.)
	Anil Keste (Alt.)
	Tony Kilwein (Alt.)
	Curtis Stevens (Alt.)
	Victor Taberski (Alt.)
	Judy Westby (Alt.)
Silicon Motion, Inc.	Amanda Huang
	Edward Hsieh (Alt.)
	Randy Hsu (Alt.)
	Sky Hsu (Alt.)
SK Hynix, Inc.	Taeho Lee
	Youngkyu Jeon (Alt.)
	Myron Loewen (Alt.)
	Jungki Noh (Alt.)
	Junho Son (Alt.)
Socionext	Masaru Suzuki
	Toru Iwata (Alt.)
	Hideyuki Kabuo (Alt.)
	Masanori Okinoi (Alt.)
TE Connectivity	Matt Schumacher
	Daniel Gorenc (Alt.)
	Tom Grzysiewicz (Alt.)
	Kyle Klinger (Alt.)
	Jeff Mason (Alt.)
	Joel Meyers (Alt.)
	Nathan Tracy (Alt.)
Teledyne LeCroy Corporation	Amit Bakshi
	Rakesh Kathiresan Natarajan (Alt.)
	Douglas Lee (Alt.)
Toshiba America Electronic Components, Inc.	Patrick Hery
	Johanna Hernandez (Alt.)
Unisys Corporation	Jeffrey Dremann
	Phil Shelton (Alt.)
Viavi Solutions, Inc.	Jason Rusch
	George Bullis (Alt.)
	Dominic Coupal (Alt.)
	Paul Gentieu (Alt.)
	Jason Mann (Alt.)
	Francisco Roy (Alt.)
	Yamini Shastry (Alt.)
VMware, Inc.	Neil MacLean
	Wenchao Cui (Alt.)
	Wenhua Liu (Alt.)
	Mike Panas (Alt.)
	Murali Rajagopal (Alt.)
	Ahmad Tawil (Alt.)

Organization Represented	*Name of Representative*
Western Digital Corporation	Dave Landsman Jorge Campello (Alt.) Xu Carter (Alt.) Marvin DeForest (Alt.) Kirill Dimitrov (Alt.) Michael Koffman (Alt.) Larry McMillian (Alt.) Chet Mercardo (Alt.) Nadesan Narenthiran (Alt.) Christopher Reed (Alt.) Yoni Shternhell (Alt.) Ralph Weber (Alt.) Eric Zornberg (Alt.)

Emeritus
William Ham
John Lohmeyer

Information technology — Small computer system interface (SCSI) —

Part 346:
Zoned Block Commands - 2 (ZBC-2)

1 Scope

This standard defines the model and command set extensions to facilitate operation of zoned block devices. The clauses in this standard, implemented in conjunction with the applicable clauses of SPC-6 and SBC-5, specify the standard command set for zoned block devices.

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.

T10/BSR INCITS 546, *SCSI Architecture Model - 6 (SAM-6)* (planned as ISO/IEC 14776-416)

T10/BSR INCITS 557, *SCSI / ATA Translation - 5 (SAT-5)* (under national consideration)

T10/BSR INCITS 566, *SCSI Primary Commands - 6 (SPC-6)* (planned as ISO/IEC 14776-456)

T10/BSR INCITS 571, *SCSI Block Commands - 5 (SBC-5)* (planned as ISO/IEC 14776-325)

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 access pattern requirements
requirements, in addition to requirements in SBC-5, that apply to read operations, verify operations, and write operations performed by the device server in a zoned block device

3.1.2 activate
cause a write pointer zone to become an active zone

3.1.3 active zone
write pointer zone that has a Zone Condition other than INACTIVE

3.1.4 additional sense code
combination of the ADDITIONAL SENSE CODE field and the ADDITIONAL SENSE CODE QUALIFIER field in sense data

Note 1 to entry: See SPC-6.

3.1.5 application client
object that is the source of SCSI commands

Note 1 to entry: See SAM-6.

3.1.6 byte
8-bit construct

3.1.7 cache
temporary data storage area that is capable of containing a subset of the logical block data stored by the logical unit and is either volatile or non-volatile

Note 1 to entry: See 4.13.

3.1.8 close zone operation
device server process that makes zone resources non-volatile, releases open zone resources, and results in the Zone Condition becoming CLOSED or EMPTY

Note 1 to entry: See 4.5.3.2.3.

3.1.9 command
request describing a unit of work to be performed by a device server

Note 1 to entry: See SAM-6.

3.1.10 command descriptor block (CDB)
structure used to communicate commands from an application client to a device server

Note 1 to entry: See SPC-6.

3.1.11 conventional zone
zone that is not associated with a write pointer

Note 1 to entry: See 4.5.2.

3.1.12 deactivate
cause a write pointer zone to become an inactive zone

3.1.13 device server
object within a logical unit (see 3.1.28) that processes SCSI commands according to the rules of command management

Note 1 to entry: See SAM-6.

3.1.14 ending LBA
highest numbered LBA of a command

3.1.15 field
group of one or more contiguous bits, a part of a larger structure

Note 1 to entry: Examples of larger structures are a CDB (see 3.1.10) and sense data (see SPC-6).

3.1.16 finish zone operation
device server process that results in the Zone Condition becoming FULL

Note 1 to entry: See 4.5.3.2.4.

3.1.17 format operation
process by which a device server initializes the medium in a logical unit

Note 1 to entry: See 4.10.

3.1.18 gap zone
zone in which all of the LBAs are incapable of storing user data

Note 1 to entry: Gap zones enable the alignment of the starting LBAs of zones that are capable of storing user data (e.g., as described in 4.2.4.1 and 4.9).

Note 2 to entry: See 4.5.4.

3.1.19 hard reset
condition resulting from the events defined by SAM-6 during which the SCSI device performs the hard reset operations described in SAM-6, this standard, and other applicable command standards

3.1.20 host aware zoned block device
logical unit that implements the host aware zoned block device model

Note 1 to entry: See 4.2.2.

3.1.21 host managed zoned block device
logical unit that implements the host managed zoned block device model

Note 1 to entry: See 4.2.3.

3.1.22 inactive zone
write pointer zone that has a Zone Condition of INACTIVE

3.1.23 I_T nexus
relationship between a SCSI initiator port and a SCSI target port

Note 1 to entry: See SAM-6.

3.1.24 logical block
set of data bytes accessed and referenced as a unit

Note 1 to entry: See SBC-5.

3.1.25 logical block access command
command that requests access to one or more logical blocks that may require access to the medium

Note 1 to entry: See SBC-5.

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

Note 1 to entry: See SBC-5.

3.1.27 logical block data
user data and protection information, if any

3.1.28 logical unit
externally addressable entity within a SCSI target device (see 3.1.48) that implements a SCSI device model

Note 1 to entry: See SAM-6.

3.1.29 manage open zone resources operation
device server process that determines the availability of open zone resources

Note 1 to entry: See 4.5.3.2.8.

3.1.30 media
plural of medium

3.1.31 medium
material that is not cache on which data is stored (e.g., a magnetic disk)

3.1.32 non-sequential write operation
write operation performed on a write pointer zone for which the lowest LBA for that operation is not equal to the write pointer for that write pointer zone

Note 1 to entry: See 4.5.3.3.2.

3.1.33 non-volatile cache
cache that retains logical block data through any power cycle

3.1.34 open zone
zone with a Zone Condition of EXPLICITLY OPENED or IMPLICITLY OPENED

3.1.35 open zone operation
device server process that results in the Zone Condition becoming EXPLICITLY OPENED

3.1.36 open zone resources
zone resources that are available only to an open zone

Note 1 to entry: See 4.5.3.1.3.

3.1.37 persistent zone resources
zone resources that persist across all condition changes or SCSI events (see SAM-6) including a power on

Note 1 to entry: For a zone in any Zone Condition other than IMPLICITLY OPENED or EXPLICITLY OPENED, all zone resources are persistent zone resources.

3.1.38 physical block
set of data bytes accessed as a unit by the device server (see SBC-5)

3.1.39 power cycle
sequence of power being removed followed by power being applied to a SCSI device

3.1.40 power on
condition resulting from the events defined by SAM-6 during which a SCSI device performs the power on operations described in SAM-6, this standard, and other applicable command standards

3.1.41 protection information
group of fields at the end of each logical block or at specified intervals within each logical block that contain a logical block guard, an application tag, and a reference tag

Note 1 to entry: See SBC-5.

3.1.42 read command
command that requests read operations

3.1.43 read operation
read operation as described in SBC-5 with the additional requirements described in this standard

3.1.44 realm
set of zone ranges, each in a different zone domain, and an indicated relationship among those zone ranges

Note 1 to entry: See 4.2.4.6.

3.1.45 realm attribute
named property associated with a realm

Note 1 to entry: In the absence of other conditions or actions (e.g., actions by the application client), realm attributes do not affect the operation of the zone condition state machine.

Note 2 to entry: See 4.4.

3.1.46 reset write pointer operation
device server process that results in the write pointer being set to the lowest LBA of the write pointer zone

Note 1 to entry: See 4.5.3.2.5.

3.1.47 sanitize operation
process by which a device server alters information on a logical unit such that recovery of previous logical block data from the cache and the medium is not possible

Note 1 to entry: See 4.11.

3.1.48 SCSI target device
SCSI device containing logical units and SCSI target ports that receives device service requests and task management requests for processing and sends device service responses and task management responses to SCSI initiator devices

Note 1 to entry: See SAM-6.

3.1.49 sense key
contents of the SENSE KEY field in the sense data

Note 1 to entry: See SPC-6.

3.1.50 sequential or before required zone
write pointer zone in which the device server allows write commands that specify a starting LBA that is less than or equal to that zone's write pointer

Note 1 to entry: See 4.5.3.5.

3.1.51 sequential write preferred zone
write pointer zone in which the device server allows write operations that specify a lowest LBA that is within the zone but is not equal to the zone's write pointer

Note 1 to entry: See 4.5.3.3.

3.1.52 sequential write required zone
write pointer zone in which the device server requires that the lowest LBA for a write operation be the LBA indicated by the zone's write pointer

Note 1 to entry: See 4.5.3.4.

3.1.53 starting LBA
lowest numbered LBA of a command

3.1.54 state machine variable
variable that exists within the context of a state machine

Note 1 to entry: A state machine variable may contain status from one state that is used in another state of the same state machine.

Note 2 to entry: The value contained in a state machine variable may affect subsequent state transitions or state machine outputs.

3.1.55 status
one byte of response information that contains a coded value defined in SAM-6, transferred from a device server to an application client upon completion of each command

Note 1 to entry: See SAM-6.

3.1.56 Trusted Computing Group
organization that develops and promotes open standards for hardware-enabled trusted computing and security technologies

Note 1 to entry: See https://www.trustedcomputinggroup.org.

3.1.57 unit attention condition
state that a logical unit (see 3.1.28) maintains while the logical unit has asynchronous status information to report to the SCSI initiator ports associated with one or more I_T nexuses (see 3.1.23)

Note 1 to entry: See SAM-6.

3.1.58 unmet prerequisite
requirement that, as a result of its not being met, prevents the successful performing of a zone activation operation (see 4.5.3.2.7)

Note 1 to entry: Unmet prerequisites are indicated by bits in the parameter data (see 5.11.3.1) returned by a command (e.g., the NOT EMPTY bit) as described in 4.5.3.2.7.2.

3.1.59 user data
data contained in logical blocks that is accessible by an application client and is neither protection information nor other information that may not be accessible to the application client

3.1.60 verify command
command that requests verify operations

3.1.61 verify operation
verify operation as described in SBC-5 with the additional requirements described in this standard

3.1.62 volatile cache
cache that does not retain logical block data between power cycles

3.1.63 write command
command that requests write operations

3.1.64 write operation
write operation as described in SBC-5 with the additional requirements described in this standard

3.1.65 write pointer
pointer to a logical block in a write pointer zone where the next write operation in that zone should start

Note 1 to entry: See 4.5.3.1.

3.1.66 write pointer zone
zone that has an associated write pointer

Note 1 to entry: See 4.5.3.

3.1.67 zone
one of a set of disjoint contiguous ranges of LBAs that comprise a logical unit

Note 1 to entry: See 4.5.

3.1.68 zone activation operation
operation that manages active zones and inactive zones

Note 1 to entry: See 4.5.3.2.7.

3.1.69 zone attribute
named property associated with a zone

Note 1 to entry: This standard assumes that all zone attributes (see 4.3) are zone resources regardless of how the device sever maintains the defined information.

Note 2 to entry: In the absence of other conditions or actions (e.g., actions by the application client), zone attributes do not affect the operation of the zone condition state machine.

Note 3 to entry: See 4.3.

3.1.70 zone domain
zone range in which all of the zones are able to be active zones at the same time

Note 1 to entry: See 4.2.4.2.

3.1.71 zone ID
lowest LBA of a zone

3.1.72 zone range
contiguous set of zones specified by a zone ID and a non-zero integer number of zones

3.1.73 zone resources
device server resources that are used by a zone

Note 1 to entry: This standard defines open zone resources and persistent zone resources.

3.1.74 zoned block device
host aware zoned block device, host managed zoned block device, or domains and realms zoned block device

Note 1 to entry: See 4.2.

3.1.75 zoned maximum address
maximum LBA as indicated by the MAXIMUM LBA field (see 5.8.2) returned by the REPORT ZONES command

Note 1 to entry: See 4.8.

3.2 Symbols and abbreviations

3.2.1 Abbreviations

Abbreviations used in this standard:

Abbreviation	Meaning
CDB	command descriptor block
LBA	logical block address
LSB	least significant bit
MSB	most significant bit
n/a	not applicable
SAM-6	SCSI Architecture Model - 6 (see clause 2)
SAT-5	SCSI / ATA Translation - 5 (see clause 2)
SBC-5	SCSI Block Commands - 5 (see clause 2)
SCSI	Small Computer System Interface family of standards
SPC-6	SCSI Primary Commands - 6 (see clause 2)
TCG	Trusted Computing Group (see 3.1.56)
VPD	Vital product data

3.2.2 Mathematical operators

Mathematical operators used in this standard:

Mathematical Operators	Meaning
^ or XOR	exclusive logical OR
x	multiplication
/	division
±	plus or minus
+	add
–	subtract
AND	logical AND
< 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

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

Note 1 to entry: Significant uses of "may or may not" occur in descriptions where attention is being drawn to the "may not" case.

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 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 an 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

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

3.3.11 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., OPERATION CODE field).

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., OFFLINE is a value of the Zone Condition zone attribute). 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:

1) text;
2) tables; and
3) figures.

Not all tables or figures are fully described in the text.

Notes and examples do not constitute any requirements.

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).

Variables (i.e., alphanumeric names that represent values in computations and other statements) are represented in the same San-serif font as other information in this standard.

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).

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).

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 data dword

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 an element dword

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 that 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 (i.e., 3.6).

3.7 Notation for state diagrams

All state diagrams use the notation shown in figure 1.

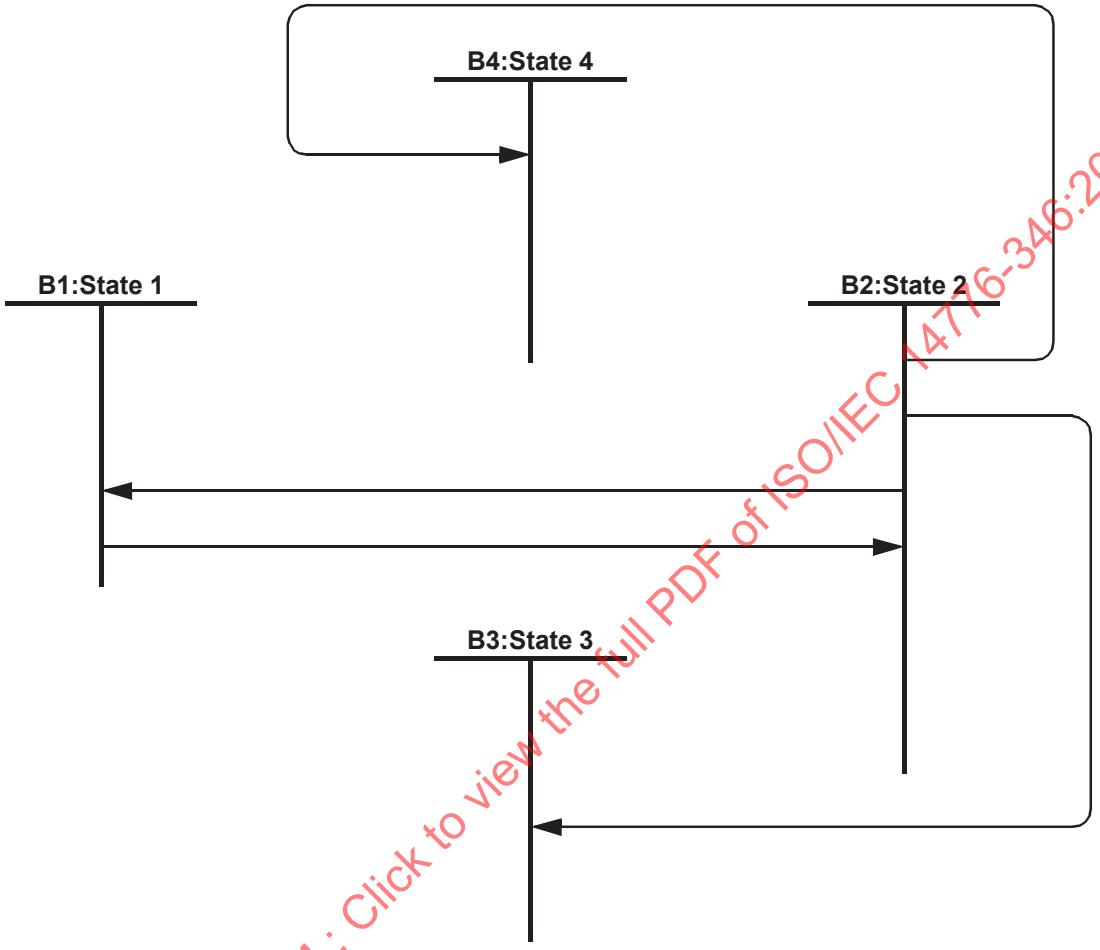

Figure 1 – Example state diagram

Each state is identified by a state designator and a state name. The state designator (e.g., B1) is unique among all state machines in this standard. The state name (e.g., State 1) is a brief description of the primary action taken during the state, and the same state name may be used by other state machines. Actions taken while in each state are described in the state description text.

Transitions between states are shown with solid lines, with an arrow pointing to the destination state.

The conditions and actions are described fully in the transition description text.

Transitions between states are instantaneous.

Upon entry into a state, all actions to be processed in that state are processed. If a state is re-entered from itself, all actions to be processed in the state are processed again. A state may be entered and exited in zero time if the conditions for exiting the state are valid upon entry into the state.

4 Zoned Block Device Model

4.1 Zoned Block Device model overview

4.1.1 Established SCSI concepts

Table 5 lists established SCSI concepts that apply to zoned block devices.

Table 5 – Zoned block device model concepts (part 1 of 2)

Concept [a]	Reference
Zoned block device models	4.2
Zone attributes	4.3
Realm attributes	4.4
Zone type models	4.5
Error reporting	4.5.3 and SBC-5
Zoned block device extensions to block device model	4.6
Capacity reporting and LBAs out of range	4.8
Constant zone starting LBA values	4.9
Format operations	4.10 and SBC-5
Sanitize operations	4.11 and SBC-5
Reservations	4.12, SPC-6, and SBC-5
Caches	4.13
Pseudo unrecovered errors	4.14 and SBC-5
Storage element depopulation and restoration operations that modify data	4.15.1 and SBC-5
Storage element depopulation with zone modifications	4.15.2 and SBC-5
Background scan operations	SBC-5
Deferred microcode activation	SBC-5
Grouping function	SBC-5
Implicit head of queue command processing	SBC-5
Initialization	SBC-5
Media examples	SBC-5
Medium defects	SBC-5

[a] Except for host managed zoned block devices, zoned block devices may support any concept or model defined in SBC-5. Host managed zoned block devices should not support SBC-5 model subclauses not listed in this table.

Table 5 – Zoned block device model concepts (part 2 of 2)

Concept [a]	Reference
Logical blocks	SBC-5
Physical blocks	SBC-5
Protection information model	SBC-5
Ready state	SBC-5
START STOP UNIT and power conditions	SBC-5
Write and unmap failures	SBC-5
Write protection	SBC-5

[a] Except for host managed zoned block devices, zoned block devices may support any concept or model defined in SBC-5. Host managed zoned block devices should not support SBC-5 model subclauses not listed in this table.

4.1.2 Peripheral device type and supported commands

Zoned block devices indicate their primary model using the PERIPHERAL DEVICE TYPE field in the standard INQUIRY data (see SPC-6). Peripheral device type of:

a) 14h indicates a host managed zoned block device (see 4.2.3); and
b) 00h (i.e., direct access block device) indicates:
   A) a host aware zoned block device (see 4.2.2); or
   B) a domains and realms zoned block device (see 4.2.4).

Application clients may use the contents of the PERIPHERAL DEVICE TYPE field as an indication of what commands may be supported by the device server, as follows:

a) 00h indicates that command support requirements are defined in SBC-5 and this standard (see table 6); and
b) 14h indicates that command support requirements are defined in this standard (see table 7).

Additional information may be available in the ZONED BLOCK DEVICE EXTENSION field in the Zoned Block Device Characteristics VPD page (see 6.5.2).

For commands that are unique to zoned block devices (see table 19), support requirements are shown in table 6.

4.2 Zoned Block Device models

4.2.1 Zoned Block Device models introduction

Zoned block device models described in this standard are as follows:

a) host aware zoned block device (see 4.2.2);
b) host managed zoned block device (see 4.2.3); and
c) domains and realms zoned block device (see 4.2.4).

Zoned block devices are accessed using LBAs. The LBAs are divided into ranges called zones (see 4.5). The entire capacity of a zoned block device is organized into a set of logically contiguous, non-overlapping zones. Figure 2 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.

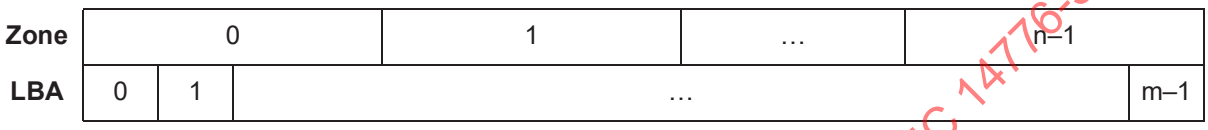

Figure 2 – Zones in a zoned block device

Each zone of a zoned block device has an associated zone type (see 4.5). Zoned block devices support the REPORT ZONES command (see 5.8) for reporting the zone type and other information for each zone.

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

Table 6 – Requirements of zoned block devices

Characteristic	Reference	Host aware zoned block device	Host managed zoned block device	Domains and realms zoned block device
Command support	4.1.2	See SPC-6 and SBC-5	See table 7	See SPC-6 and SBC-5
PERIPHERAL DEVICE TYPE field value	4.1.2	00h	14h	00h
ZONED BLOCK DEVICE EXTENSION field value in the Zoned Block Device Characteristics VPD page	6.5.2	1h	0h	2h
Conventional zone	4.5.2	Optional	Optional	Not supported
Sequential write preferred zone	4.5.3.3	Mandatory	Not supported	Optional [a]
Sequential write required zone	4.5.3.4	Not supported	Mandatory	Optional [a]
Sequential or before required zone	4.5.3.5	Not supported	Not supported	Mandatory
Gap zone	4.5.4	Not supported	see 4.2.3	Optional
CLOSE ZONE command	5.2	Mandatory	Mandatory	Mandatory
FINISH ZONE command	5.3	Mandatory	Mandatory	Mandatory
OPEN ZONE command	5.4	Mandatory	Mandatory	Mandatory
REMOVE ELEMENT AND MODIFY ZONES command	5.5	Optional	Optional	Optional
REPORT REALMS command	5.6	Not supported	Not supported	Optional
REPORT ZONE DOMAINS command	5.7	Not supported	Not supported	Mandatory
REPORT ZONES command	5.8	Mandatory	Mandatory	Mandatory
RESET WRITE POINTER command	5.9	Mandatory	Mandatory	Mandatory
SEQUENTIALIZE ZONE command	5.10	Mandatory	Not supported	Optional [b]
ZONE ACTIVATE command	5.11	Not supported	Not supported	Mandatory
ZONE QUERY command	5.12	Not supported	Not supported	Mandatory

[a] The device server shall support one or more of the following types of zones:
a) sequential write preferred zones; or
b) sequential write required zones.

[b] If the device server supports sequential write preferred zones (see 4.5.3.3), then the SEQUENTIALIZE ZONE command (see 5.10) is 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) (see SPC-6); and
b) the ZONED BLOCK DEVICE EXTENSION field set to 1h in the Zoned Block Device Characteristics VPD page (see 6.5.2).

Host aware zoned block devices:

a) may support conventional zones (see 4.5.2);
b) shall support one or more sequential write preferred zones (see 4.5.3.3);
c) shall not support sequential write required zones (see 4.5.3.4);
d) shall not support sequential or before required zones (see 4.5.3.5); and
e) shall not support gap zones (see 4.5.4).

Capacity reporting requirements are defined in 4.8.

In addition to primary commands (see SPC-6) and commands for direct access block devices (see SBC-5), host aware zoned block devices shall support the commands specified in table 6.

4.2.3 Host managed zoned block device model

A host managed zoned block device reports:

a) the PERIPHERAL DEVICE TYPE field set to 14h (i.e., host managed zoned block device) (see SPC-6); and
b) the ZONED BLOCK DEVICE EXTENSION field set to 0h in the Zoned Block Device Characteristics VPD page (see 6.5.2).

Host managed zoned block devices:

a) may support conventional zones (see 4.5.2);
b) shall not support sequential write preferred zones (see 4.5.3.3);
c) shall support one or more sequential write required zones (see 4.5.3.4); and
d) shall not support sequential or before required zones (see 4.5.3.5).

If the ZONE ALIGNMENT METHOD field (see 6.5.2) is:

a) set to 8h, then the host managed zone block device may support gap zones (see 4.5.4) as described in 4.9; and
b) not set to 8h, then the host managed zone block device shall not support gap zones.

Capacity reporting requirements are defined in 4.8.

Table 7 lists the commands for host managed zoned block devices.

Table 7 – Commands for host managed zoned block devices (part 1 of 2)

Command	Operation code	Type	LBACT	Reference
ATA PASS-THROUGH (12)	A1h	O	n/a	SAT-5
ATA PASS-THROUGH (16)	85h	O	n/a	SAT-5
CLOSE ZONE	94h/01h	M	Z	5.2
FINISH ZONE	94h/02h	M	Z	5.3
FORMAT UNIT	04h	O	Z	SBC-5 and 4.10
FORMAT WITH PRESET	38h	O	Z	SBC-5 and 4.10
INQUIRY	12h	M	n/a	SPC-6
LOG SELECT	4Ch	O	n/a	SPC-6
LOG SENSE	4Dh	M	n/a	SPC-6
MODE SELECT (10)	55h	M	n/a	SPC-6
MODE SENSE (10)	5Ah	M	n/a	SPC-6
OPEN ZONE	94h/03h	M	Z	5.4
PERSISTENT RESERVE IN	5Eh	O	n/a	SPC-6
PERSISTENT RESERVE OUT	5Fh	O	n/a	SPC-6
READ (16)	88h	M	R	SBC-5
READ BUFFER (10)	3C	O	n/a	SPC-6
READ BUFFER (16)	9Bh	O	n/a	SPC-6
READ CAPACITY (16)	9Eh/10h	M	n/a	SBC-5 and 4.8
READ DEFECT DATA (12)	B7h	O	n/a	SBC-5
REMOVE ELEMENT AND MODIFY ZONES	9Eh/1Ah	O	Z	5.5
REMOVE ELEMENT AND TRUNCATE	9Eh/18h	O	Z	SBC-5
REPORT LUNS	A0h	M	n/a	SPC-6
REPORT SUPPORTED OPERATION CODES	A3h/0Ch	M	n/a	SPC-6

Key:
O = optional
M = mandatory
R = read command
U = unmap command
V = verify command
W = write command
Z = other command
LBACT = logical block access command type (see SBC-5)

Table 7 – Commands for host managed zoned block devices (part 2 of 2)

Command	Operation code	Type	LBACT	Reference
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	A3h/0Dh	M	n/a	SPC-6
REPORT TIMESTAMP	A3/0Fh	O	n/a	SPC-6
REPORT ZONES	95h/00h	M	Z	5.8
REQUEST SENSE	03h	M	n/a	SPC-6
RESET WRITE POINTER	94h/04h	M	Z	5.9
RESTORE ELEMENT AND REBUILD	9Eh/19h	O	Z	SBC-5
SANITIZE	48h	O	Z	SBC-5 and 4.11
SECURITY PROTOCOL IN	A2h	O	n/a	SPC-6
SECURITY PROTOCOL OUT	B5h	O	n/a	SPC-6
SEND DIAGNOSTIC	1Dh	O	n/a	SPC-6
SET TIMESTAMP	A4h/0Fh	O	n/a	SPC-6
START STOP UNIT	1Bh	M	n/a	SBC-5
SYNCHRONIZE CACHE (16)	91h	M	W	SBC-5
TEST UNIT READY	00h	M	n/a	SPC-6
VERIFY (16)	8Fh	O	V, W	SBC-5
WRITE (16)	8Ah	M	W	SBC-5
WRITE BUFFER	3Bh	O	n/a	SPC-6
WRITE LONG (16)	9Fh/11h	O	Z	SBC-5 and 4.14
WRITE SAME (16)	93h	M	U, W	SBC-5

Key:
O = optional
M = mandatory
R = read command
U = unmap command
V = verify command
W = write command
Z = other command
LBACT = logical block access command type (see SBC-5)

4.2.4 Domains and realms zoned block device model

4.2.4.1 Domains and realms zoned block device model overview

A domains and realms zoned block device reports:

a) the PERIPHERAL DEVICE TYPE field set to 00h (i.e., direct access block device) (see SPC-6); and
b) the ZONED BLOCK DEVICE EXTENSION field set to 2h in the Zoned Block Device Characteristics VPD page (see 6.5.2).

Domains and realms zoned block devices:

a) shall not support conventional zones (see 4.5.2);
b) shall support one or more of the following types of zones:
   A) sequential write preferred zones (see 4.5.3.3); or
   B) sequential write required zones (see 4.5.3.4);
c) shall support one or more sequential or before required zones (see 4.5.3.5); and
d) may support gap zones (see 4.5.4).

Capacity reporting requirements are defined in 4.8.

In addition to primary commands (see SPC-6) and commands for direct access block devices (see SBC-5), domains and realms zoned block devices shall support the commands specified in table 6.

Zone domains and realms enhance the basic zoned block device model (see 4.2.1) to define a mechanism that allows an application client to manage contention for certain SCSI device resources in a way that is independent of many details about those SCSI device resources. Two or more zone domains (see 4.2.4.2) represent the SCSI device resources for which an application client is able to manage contention.

The SCSI device resources for which contention management is provided include, but are not limited to, the ability to store logical block data. The contention for SCSI device resources is managed by having:

a) some zones be active zones (i.e., able to use the SCSI device resource in contention); and
b) the remaining zones be inactive zones (i.e. unable to use the SCSI device resource in contention).

The REPORT ZONES command (see 5.8), REPORT REALMS command (see 5.6), REPORT ZONE DOMAINS command (see 5.7), and the ZONE QUERY command (see 5.12) provides current indications of what zone domains and zone types are available for the application client to choose to make active zones.

The ZONE ACTIVATE command (see 5.11) may activate a zone range in one zone domain and may deactivate a corresponding zone range in one or more different zone domains.

Commands that specify LBAs (e.g., read commands and write commands) in inactive zones are processed as described in 4.5.3.1.5 and 4.5.3.1.6.

4.2.4.2 Zone domains

A zone domain (see figure 3) is a zone range that is identified by a zone domain ID (see table 8). A zone domain ID is assigned by the device server starting at zero and incrementing by one.The zone range that constitutes the Zone domain shown in figure 3:

1) starts with Zone t, followed by Zone t+1, Zone t+2, and Zone t+3;
2) continues for zero or more zones; and
3) ends with Zone x-3, followed by Zone x-2, Zone x-1 and Zone x.

Zone t is the first zone in the Zone domain shown in figure 3 and the lowest LBA in Zone t is the lowest LBA of the Zone domain. Zone x is the last zone in the Zone domain shown in figure 3 and the highest LBA in Zone x is the highest LBA of the Zone domain.

Zone domain

Lowest LBA	Zone t	Zone t+1	Zone t+2	Zone t+3	• • •	Zone x-3	Zone x-2	Zone x-1	Zone x	Highest LBA

Figure 3 – Zone domain

Table 8 – Zone domain ID values

Code	Description
00h to F7h	Zone domain ID values
F8h to FEh	Reserved
FFh	No zone domain ID specified

All the zones in a single zone domain shall be capable of being active zones at the same time.

Each zone that is capable of being activated shall be included in one and only one zone domain.

Two different zone domains may have:

a) different numbers of zones; and
b) different numbers of LBAs.

Figure 4 shows an example of a device server that supports two zone domains with:

1) Zone domain 0:
   A) starting with Zone a and ending with Zone b; and
   B) containing:
      a) LBA 0 as the lowest LBA in Zone domain 0; and
      b) LBA m as the highest LBA in Zone domain 0;
2) zero or more gap zones (see 4.5.4) that separate Zone domain 0 from Zone domain 1, with
   A) LBA m+1 as the lowest LBA of the lowest gap zone, if any; and
   B) LBA p-1 as the highest LBA of the highest gap zone, if any;
   and
3) Zone domain 1:
   A) starting with Zone y and ending with Zone z; and
   B) containing:
      a) LBA p as the lowest LBA in Zone domain 1; and
      b) LBA r as the highest LBA in Zone domain 1.

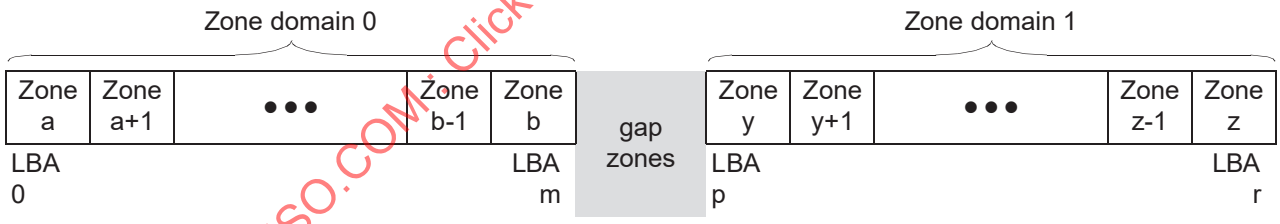

Figure 4 – Example of two zone domains

4.2.4.3 Zone domain 0

A domains and realms zoned block device shall have one zone domain with LBA 0 as its lowest numbered LBA. That zone domain is identified as zone domain 0.

As described in 4.8, the highest LBA in zone domain 0 establishes the value returned in the RETURNED LOGICAL BLOCK ADDRESS field in the READ CAPACITY (16) parameter data.

At the time of manufacture, a domains and realms zoned block device shall have a zone domain 0 that contains only sequential or before required zones (see 4.5.3.5) that have a Zone Condition of FULL.

4.2.4.4 Zone domains other than zone domain 0

For zone domains other than zone domain 0, the device server shall support accesses (e.g., reads and writes) that specify LBAs that are greater than the contents of the RETURNED LOGICAL BLOCK ADDRESS field in the READ CAPACITY (16) parameter data (see SBC-5 and 4.8).

The LBAs in the zones in zone domains other than zone domain 0 are:

a) greater than the value in the RETURNED LOGICAL BLOCK ADDRESS field; and
b) less than or equal to the zoned maximum address.

4.2.4.5 Zone activation

A successful ZONE ACTIVATE command (see 5.11) that specifies a zone range to be activated activates one or more zones and deactivates zero or more zones. A successful ZONE ACTIVATE command that specifies a zone range to be deactivated activates zero or more zones and deactivates one or more zones.

The smallest zone range processed by a ZONE ACTIVIATE command or a ZONE QUERY command (see 5.12) is:

a) one zone, if the AAORB bit (see 6.5.2) is set to zero (see example in figure 5); and
b) all the zones in one zone realm, if the AAORB bit is set to one (see example in figure 8).

Figure 5 shows an example of the before and after conditions for a ZONE ACTIVATE command that is processed with the AAORB bit set to zero.

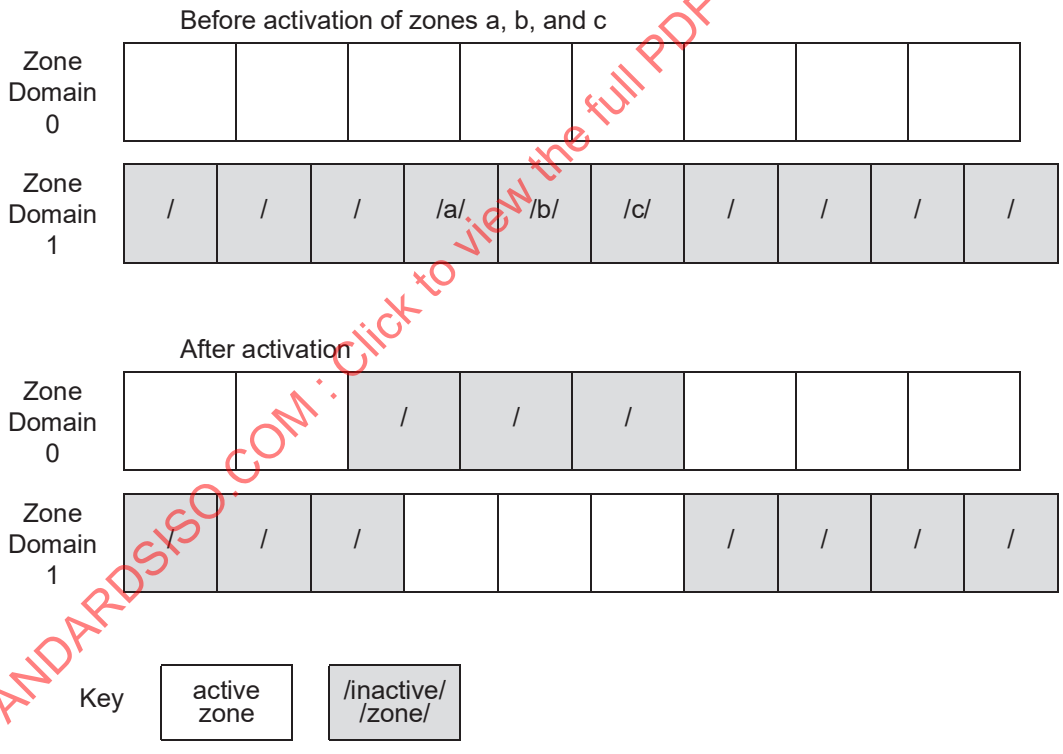

Figure 5 – Example of zone activation with the AAORB bit set to zero

4.2.4.6 Realms

Combined with zone domains (see 4.2.4.2), realms form a matrix (see figure 6) that provides applications clients with a set of choices for which zones are active. In figure 6, zone domain 0 contains LBA 0 (see 4.2.4.3) and each realm contains more than one zone. Figure 3 describes how zones are contained within zone domains.

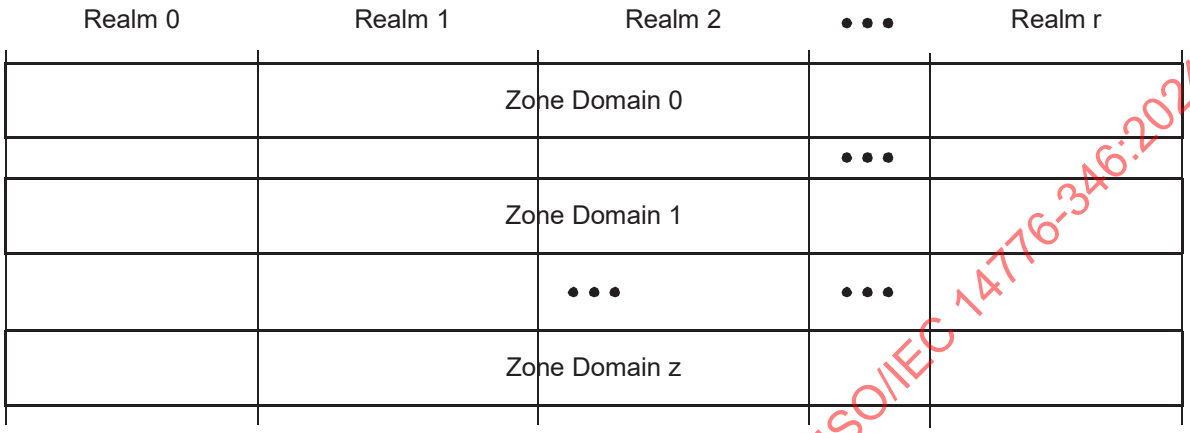

Figure 6 – Realms model

Each realm shall contain at least one zone.

Realms associate characteristics with:

a) zone domains (e.g., zone type, recording technology, data transfer performance); and
b) realms (e.g., location on the media, data storage density, storage density performance).

EXAMPLE - Figure 7 shows a configuration in which one zone domain uses conventional recording in sequential or before required zones (see 4.5.3.5) and one zone domain uses shingled magnetic recording in sequential write required zones (see 4.5.3.4). In this configuration, the domains and realms zoned block device allows a host to modify which realms store logical block data without the full effects of shingled magnetic recording and which realms store logical block data using shingled magnetic recording.

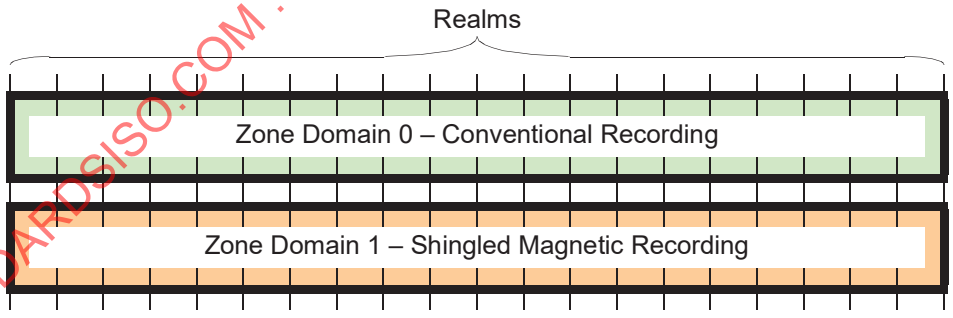

Figure 7 – Example of conventional and shingled recording technologies using two zone domains

The ZONE ACTIVATE command (see 5.11) modifies which zones are active zones within a contiguous set of realms.

The result of processing a ZONE ACTIVATE command is that zones in one zone domain within a realm become active zones at the same time as zones in a different zone domain within the same realm become inactive zones.

At any point in time, all the zones in one realm are active zones in only one of the zone domains associated with that realm.

Figure 8 shows an example of the before and after conditions for a ZONE ACTIVATE command that is processed with the AAORB bit set to one (i.e., in the case where realm boundaries affect the processing of a ZONE ACTIVATE command).

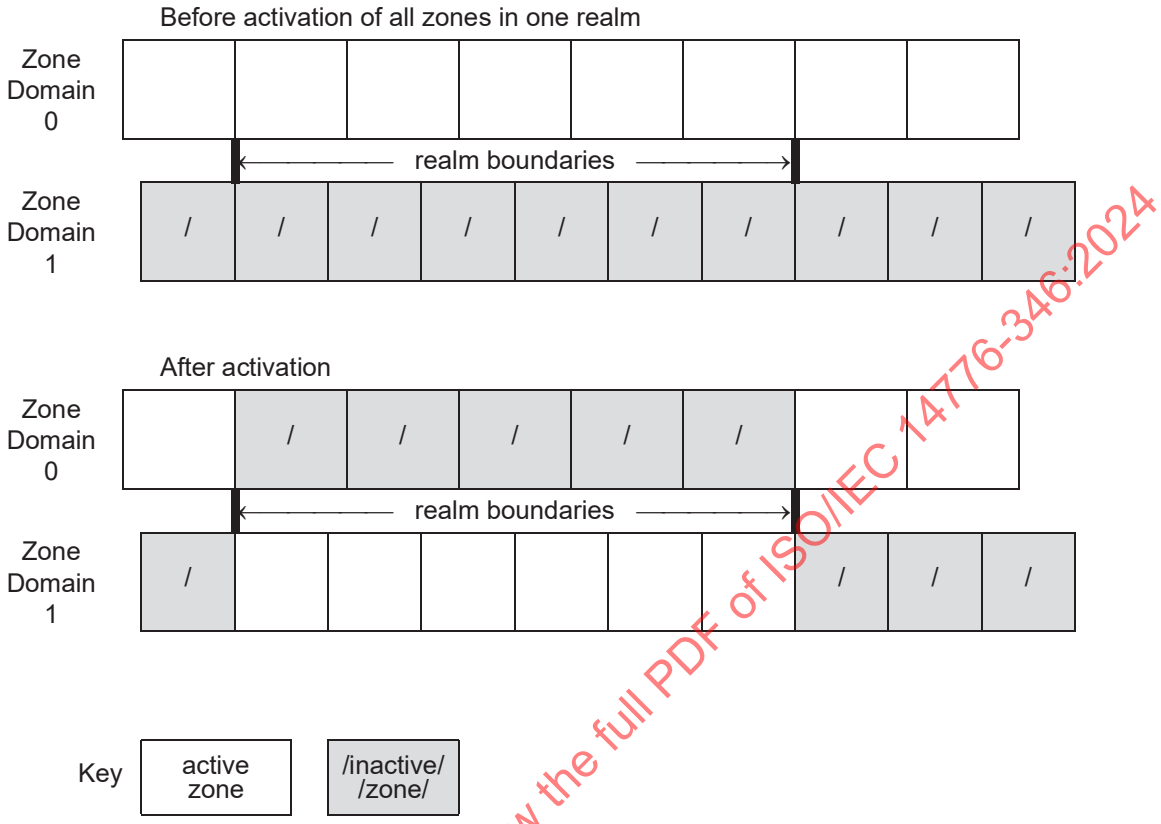

Figure 8 – Example of zone activation with the AAORB bit set to one

4.2.4.7 Realm boundary considerations

If the SRB bit (see 5.7.2) is set to zero for a processed zone domain, then the processing of a ZONE ACTIVATE command (see 5.11) shall not result in any realm boundaries changing in that zone domain. If the SRB bit is set to one for a processed zone domain, then updated realm boundary information is available from the REPORT REALMS command (see 5.6) after a ZONE ACTIVATE command is processed to successful completion.

Changes in realm boundary information may result in the RLM ALIGN bit (see 5.11.3) being set to one as an unmet prerequisite for a subsequent ZONE ACTIVATE command, unless the updated information provided by a REPORT REALMS command is used to prepare that ZONE ACTIVATE command.

The realm starting LBAs and ending LBAs returned by a REPORT REALMS command shall not change unless the device server processes a ZONE ACTIVATE command, a FORMAT UNIT command (see SBC-5), or a FORMAT WITH PRESET command (see SBC-5).

4.3 Zone attributes

4.3.1 Zone attributes summary

The zoned block device zone attributes are summarized in table 9.

Table 9 – Summary of zone attributes

Zone attribute	Description	Reference
Zone Type	Type of zone (e.g., CONVENTIONAL)	4.3.2
Zone Condition	Zone's operational characteristics	4.3.3
WPointer	Combination of the write pointer (see 4.5.3.1) and whether the write pointer is valid	4.3.4
RWP Recommended	Indication of whether the processing of a RESET WRITE POINTER command (see 5.9) for this zone is recommended	4.3.5
Non-Sequential Write Resources Active	Indication of whether zone resources are allocated for non-sequential write operations	4.3.6
Predicted Unrecovered Errors Present	Indication of whether predicted unrecovered errors (see SBC-5) are associated with logical blocks in the zone	4.3.7

4.3.2 Zone Type zone attribute

The Zone Type zone attribute (see table 10) indicates the type of zone.

Table 10 – Zone Type zone attribute

Zone Type	Reference
CONVENTIONAL	4.5.2
SEQUENTIAL WRITE PREFERRED	4.5.3.3
SEQUENTIAL WRITE REQUIRED	4.5.3.4
SEQUENTIAL OR BEFORE REQUIRED	4.5.3.5
GAP	4.5.4

The relationships between Zone Type and other zone attributes are shown in table 11.

Table 11 – Relationships between zone attributes

Zone Type	Other zone attributes			
	Non-Sequential Write Resources Active	RWP Recommended	WPointer	Zone Condition
Write pointer zones (see 4.5.3)				
SEQUENTIAL WRITE PREFERRED	see 4.3.6	see 4.3.5	see 4.5.3.1 and table 15	see 4.3.3
SEQUENTIAL WRITE REQUIRED	false [a]			
SEQUENTIAL OR BEFORE REQUIRED	false [b]			
Other zone types				
CONVENTIONAL	false [c]	false [c]	invalid [c]	NOT WRITE POINTER
GAP	false	false	invalid	

[a] Zone resources for non-sequential write operations are not allocated for a zone in which all non-sequential write commands are terminated with an error.
[b] For a sequential or before required zone (see 4.5.3.5), the values of zone attributes are required to be compatible with the direct access block device type defined in SBC-5.
[c] For a conventional zone (see 4.5.2), the values of zone attributes are required to be compatible with the direct access block device type defined in SBC-5.

4.3.3 Zone Condition zone attribute

The Zone Condition zone attribute (see table 12) is determined by Zone Type (see 4.3.2) and the zone condition state machine state (see 4.5.3.6).

Table 12 – Zone Condition zone attribute

Zone Type	Zone condition state machine	Reference	Zone Condition
Write pointer zones (see 4.5.3)			
SEQUENTIAL WRITE PREFERRED or SEQUENTIAL WRITE REQUIRED	ZC1:Empty state	4.5.3.6.2	EMPTY
	ZC2:Implicit_Open state	4.5.3.6.3	IMPLICITLY OPENED
	ZC3:Explicit_Open state	4.5.3.6.4	EXPLICITLY OPENED
	ZC4:Closed state	4.5.3.6.5	CLOSED
	ZC5:Full state	4.5.3.6.6	FULL
	ZC6:Read_Only state	4.5.3.6.7	READ ONLY
	ZC7:Offline state	4.5.3.6.8	OFFLINE
	ZC8:Inactive	4.5.3.6.9	INACTIVE
SEQUENTIAL OR BEFORE REQUIRED	ZC1:Empty state	4.5.3.6.2	EMPTY
	ZC2:Implicit_Open state	4.5.3.6.3	IMPLICITLY OPENED
	ZC5:Full state	4.5.3.6.6	FULL
	ZC6:Read_Only state	4.5.3.6.7	READ ONLY
	ZC7:Offline state	4.5.3.6.8	OFFLINE
	ZC8:Inactive	4.5.3.6.9	INACTIVE
Other zone types			
CONVENTIONAL	outside the scope of this standard	4.5.2	NOT WRITE POINTER
GAP	outside the scope of this standard	4.5.4	

4.3.4 WPointer zone attribute

For write pointer zones (see 4.5.3), the WPointer zone attribute is the combination of the write pointer (see 4.5.3.1) and whether the write pointer is valid (see table 15).

If Zone Type is CONVENTIONAL or GAP, then WPointer shall be invalid.

4.3.5 RWP Recommended zone attribute

For write pointer zones (see 4.5.3), if RWP Recommended is:

a) true, then the device server has determined (e.g., through the detection by the device server that the zone is using too many resources) that the processing of a RESET WRITE POINTER command (see 5.9) specifying this zone is recommended; or

b) false, then the device server has no recommendation for or against the processing of a RESET WRITE POINTER command specifying this zone.

RWP Recommended shall be set to false if:

a) the Zone Type is CONVENTIONAL or GAP; or
b) the Zone Condition is EMPTY, READ ONLY, OFFLINE, or INACTIVE.

If RWP Recommended changes from false to true then the device server should establish a unit attention condition for the SCSI initiator port (see SAM-6) associated with each I_T nexus with additional sense code set to ZONE RESET WRITE POINTER RECOMMENDED (see SPC-6) and the INFORMATION field (see SPC-6) set to the zone ID of the zone for which RWP Recommended was set to true.

4.3.6 Non-Sequential Write Resources Active zone attribute

For sequential write preferred zones (see 4.5.3.3), if Non-Sequential Write Resources Active is:

a) true, then the device server has zone resources allocated for the processing of non-sequential write operations (see 4.5.3.3.2) in this zone; or
b) false, then the device server does not have zone resources allocated for the processing of non-sequential write operations in this zone.

If a non-sequential write operation is performed in a zone, then the device server may set Non-Sequential Write Resources Active to true. The device server may set Non-Sequential Resources Active to false at any time if:

a) the write pointer does not indicate the lowest LBA in the zone; and
b) zone resources are no longer allocated for the processing of non-sequential write operations in this zone.

If the write pointer indicates the lowest LBA in the zone and the device server has performed any non-sequential write operations in a zone since the last time the Zone Condition was EMPTY, then Non-Sequential Resources Active shall not be set to false until the write pointer does not indicate the lowest LBA in the zone.

Non-Sequential Write Resources Active shall be set to false if:

a) the Zone Type is CONVENTIONAL, GAP, SEQUENTIAL WRITE REQUIRED, or SEQUENTIAL OR BEFORE REQUIRED;
b) the Zone Condition is EMPTY, READ ONLY, OFFLINE, or INACTIVE; or
c) the device server has not performed a non-sequential write operation in a zone since the last time the Zone Condition was EMPTY.

4.3.7 Predicted Unrecovered Errors Present zone attribute

For conventional zones (see 4.5.2), if Predicted Unrecovered Errors Present is:

a) true, then the zone contains one or more logical blocks for which:
   A) read command accesses result in predicted unrecovered read errors (see 4.15.2.3.2); and
   B) write command accesses result in predicted unrecovered write errors (see 4.15.2.3.3);
   or
b) false, then none of the logical blocks in the zone are affected by predicted unrecovered errors.

Predicted Unrecovered Errors Present may be changed to:

a) true by the processing of a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); and
b) false by the processing of a RESTORE ELEMENTS AND REBUILD command (see SBC-5).

If Predicted Unrecovered Errors Present is true, then read commands and write commands may report predicted unrecovered errors in ranges of logical blocks as described in SBC-5.

Predicted Unrecovered Errors Present shall be set to false if the Zone Type is SEQUENTIAL WRITE REQUIRED, SEQUENTIAL WRITE PREFERRED, SEQUENTIAL OR BEFORE REQUIRED, or GAP.

4.4 Realm attributes

4.4.1 Realm attributes overview

The realm attributes are summarized in table 13.

Table 13 – Summary of realm attributes

Realm attribute	Description	Reference
Restrict Write Pointer Reset	Indication of whether the processing of a RESET WRITE POINTER command (see 5.9) is restricted for all zones in this realm	4.4.2
Restrict Zone Activate	Indication of whether the processing of a ZONE ACTIVATE command (see 5.11) is restricted for all zones in this realm	4.4.3

4.4.2 Restrict Write Pointer Reset realm attribute

If the Restrict Write Pointer Reset realm attribute is:

a) true, then a RESET WRITE POINTER command (see 5.9) with the ALL bit set to:
   A) zero and a zone range that includes any zones in this realm is terminated with CHECK CONDITION status, the sense code set to ILLEGAL REQUEST, and the additional sense code set to RESET WRITE POINTER NOT ALLOWED; or
   B) one does not perform a Reset Write Pointer operation (see 4.5.3.2.5) for any zone in this realm; or
b) false, then the processing of a RESET WRITER POINTER command is not affected by the Restrict Write Pointer Reset realm attribute.

A command that initializes the media and performs a Reset Write Pointer operation (e.g., a FORMAT UNIT command, or SANITIZE command) shall not perform a Reset Write Pointer operation for any zone in a realm for which the Restrict Write Pointer Reset realm attribute is true.

4.4.3 Restrict Zone Activate realm attribute

If the Restrict Zone Activate realm attribute is:

a) true, then a ZONE ACTIVATE command (see 5.11) does not activate or deactivate any zones in this realm; or
b) false, then the processing of a ZONE ACTIVATE command is not affected by the Restrict Zone Activate realm attribute.

4.5 Zone type models

4.5.1 Zone type models overview

In a zoned block device, each zone:

a) has the zone attributes defined in 4.3; and
b) is one of the following zone types:
   A) a conventional zone (see 4.5.2);
   B) a gap zone (see 4.5.4); or
   C) a write pointer zone (see 4.5.3) that is either:
      a) a sequential write preferred zone (see 4.5.3.3);
      b) a sequential write required zone (see 4.5.3.4); or
      c) a sequential or before required zone (see 4.5.3.5).

The coded values that indicate the zone type are shown in table 43.

4.5.2 Conventional zone model

4.5.2.1 Conventional zone model overview

A conventional zone is a type of zone that is not associated with a write pointer and for which the device server performs operations as described in 4.5.2.2, 4.5.2.3, and SBC-5.

4.5.2.2 Write access pattern requirements for conventional zones

For conventional zones, if the device server processes a write command for that zone, then the device server shall process that command as described in SBC-5.

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, if a write command specifies:

a) a starting LBA that is in a conventional zone; and
b) access to a logical block that is in a different type of zone.

4.5.2.3 Read access pattern requirements for conventional zones

For conventional zones, 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 a read command or a verify command specifies:

a) a starting LBA that is in a conventional zone; and
b) access to a logical block that is in a different type of zone.

4.5.3 Write pointer zone models

4.5.3.1 Features common to all write pointer zones

4.5.3.1.1 Write pointer features

A write pointer zone is a type of zone that maintains a write pointer that indicates a location within that zone. The write pointer may be modified as a result of processing of a write command. Depending on the state of a zone (see 4.5.3.6), the write pointer for that zone may or may not be valid.

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 the next write command to that zone. The write pointer may be set to the lowest LBA of a zone by a reset write pointer operation (see 4.5.3.2.5) for that zone.

Each write pointer zone is in one of the states defined by the Zone Condition state machine (see 4.5.3.6). The state of each zone is indicated by the Zone Condition (see 4.3.3) and affects the contents of the ZONE CONDITION field in the data returned by the REPORT ZONES EXT command (see 5.8).

Read commands that complete without an error do not affect the write pointer.

Figure 9 shows a write pointer zone where no LBAs have been written since the last reset write pointer operation for that zone.

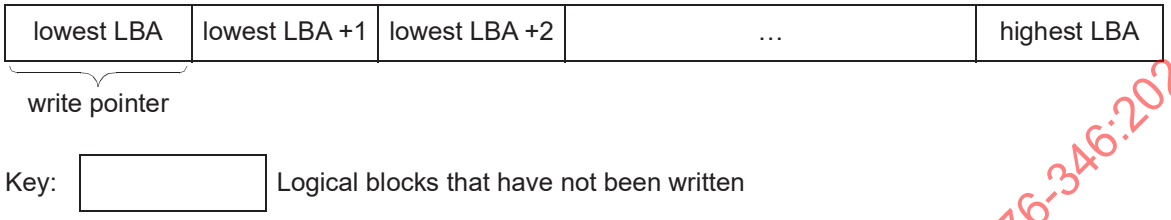

Figure 9 – Write pointer zone and write pointer after reset write pointer operation with no subsequent writes

Figure 10 shows a write pointer zone where LBAs have:

a) been written since the most recent time the zone's Zone Condition was EMPTY (i.e., LBAs less than the write pointer value); and
b) not been written since the most recent write command in this write pointer zone (i.e., LBAs greater than or equal to the write pointer value).

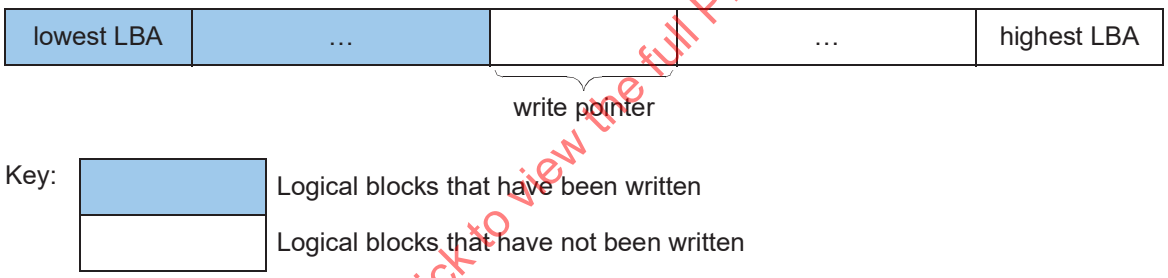

Figure 10 – Write pointer zone and write pointer

The example in figure 11 shows the effects of application client actions on the write pointer as follows:

1) at time 1, the Zone Condition is EMPTY (i.e., the write pointer is set to the lowest LBA of that zone);
2) at time 2:
   1) the Zone Condition becomes IMPLICITLY OPEN;
   2) a write operation writes logical block data starting at that zone's lowest LBA; and
   3) the write pointer is set to the highest LBA written by the write operation plus one;
3) at time 3, a write operation:
   1) writes logical block data that starts at the write pointer after time 2 and extends to the end of that zone;
   2) as a result, the Zone Condition becomes FULL; and
   3) changes the write pointer to invalid;

   and
4) at time 4:
   1) a reset write pointer operation sets the write pointer to the lowest LBA of that write pointer zone; and
   2) the Zone Condition returns to being EMPTY.

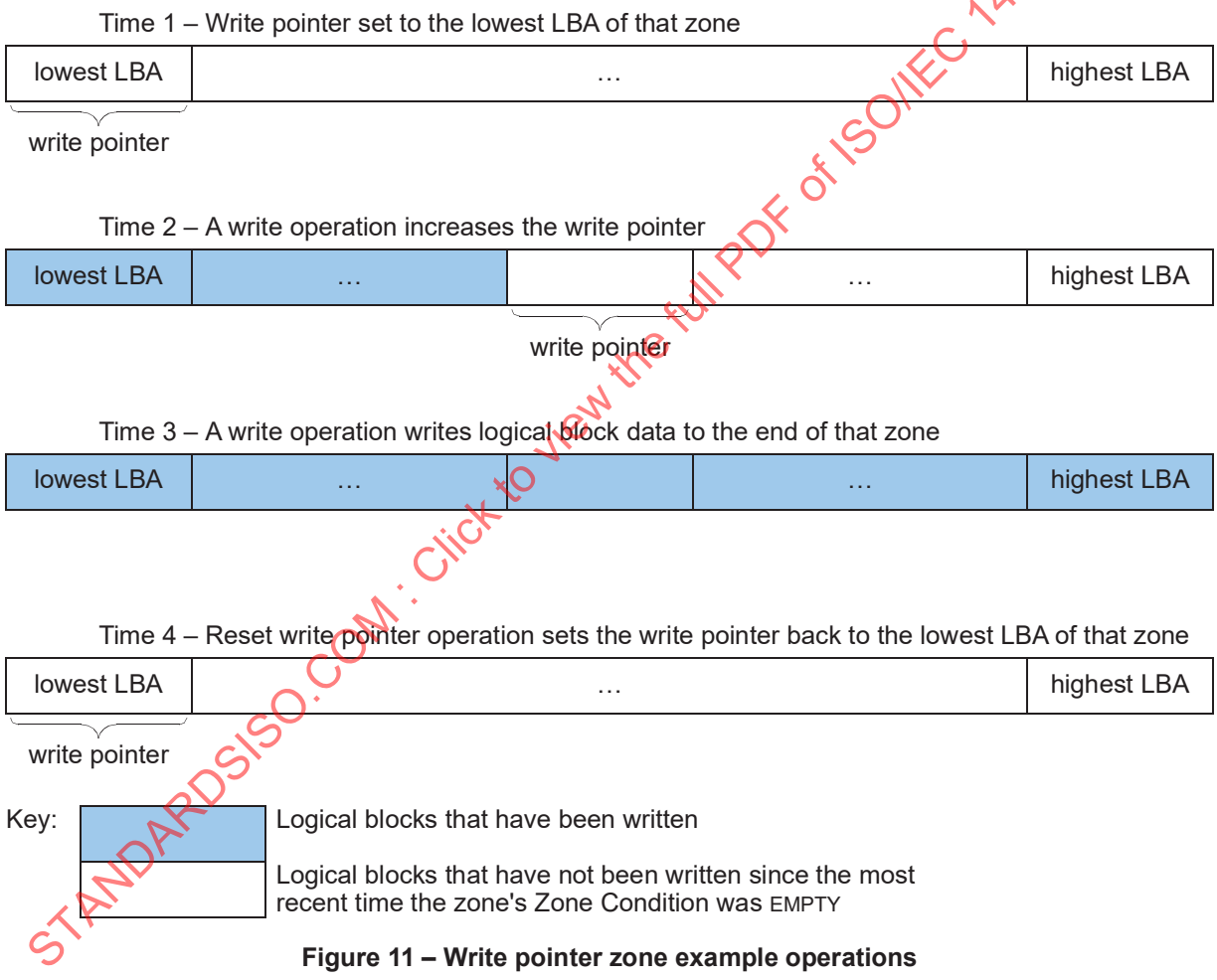

Figure 11 – Write pointer zone example operations

4.5.3.1.2 Resetting the write pointer

A reset write pointer operation (see 4.5.3.2.5) sets the write pointer in a write pointer zone to the lowest LBA of that zone and sets that zone's Zone Condition to EMPTY. A reset write pointer operation:

a) may be requested by a SANITIZE command (see 4.11 and SBC-5); and
b) is requested by:
   A) a RESET WRITE POINTER command (see 5.9);
   B) a FORMAT UNIT command (see 4.10 and SBC-5); and
   C) a FORMAT WITH PRESET command (see 4.10 and SBC-5).

4.5.3.1.3 Open zone resources

Open zone resources are zone resources that are available only to an open zone. The usage of open zone resources is managed by the zone condition state machine (see 4.5.3.6) as described for the Zone Conditions EXPLICITLY OPENED and IMPLICITLY OPENED. Open zone resources may be volatile (e.g., open zone resources may be cached copies of persistent zone resources).

The scarcity of open zone resources is associated with zone type, and this scarcity affects how those open zone resources are managed. The open zone resources for:

a) sequential write required zones (see 4.5.3.4) limit the number of sequential write required zones with a Zone Condition of EXPLICITLY OPENED or IMPLICITLY OPENED allowed at any specific time, with the limit indicated by the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field in the Zoned Block Device Characteristics VPD page (see 6.5.2);
b) sequential write preferred zones (see 4.5.3.3) have an optimal number of sequential write preferred zones with a Zone Condition of EXPLICITLY OPENED or IMPLICITLY OPENED allowed at any specific time, with the limit indicated by the OPTIMAL NUMBER OF OPEN SEQUENTIAL WRITE PREFERRED ZONES field in the Zoned Block Device Characteristics VPD page; and
c) sequential or before required zones (see 4.5.3.5) have no limits on the number of sequential or before required zones with a Zone Condition of IMPLICITLY OPENED.

The availability of open zone resources is determined by the manage open zone resources operation (see 4.5.3.2.8).

4.5.3.1.4 Initialization pattern

The initialization pattern used by a zoned block device is the logical block data set:

a) as the default initialization pattern at the time of manufacture;
b) by the most recent FORMAT UNIT command (see SBC-5), if any; or
c) by the most recent SANITIZE command with the service action set to OVERWRITE (see SBC-5), if any.

4.5.3.1.5 Write access pattern requirements common to all write pointer zones

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, if a write command specifies:

a) a starting LBA that is in a write pointer zone; and
b) access to a logical block that is in a different type of zone.

If a write command specifies the writing of one or more LBAs that are in a write pointer zone that has a Zone Condition of READ ONLY, OFFLINE, or INACTIVE, then the device server shall terminate that command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to:

a) ZONE IS READ ONLY, if that Zone Condition is READ ONLY;
b) ZONE IS OFFLINE, if that Zone Condition is OFFLINE; and
c) ZONE IS INACTIVE, if that Zone Condition is INACTIVE.

Write access pattern requirements that are specific to the WRITE LONG command (see SBC-5) are described in 4.14.

4.5.3.1.6 Read access pattern requirements common to all write pointer zones

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 a read command or a verify command specifies:

a) a starting LBA that is in a write pointer zone; and
b) access to a logical block that is in a different type of zone.

If a read command or a verify command specifies the reading of any LBA that is in a write pointer zone that has a Zone Condition of INACTIVE and the URSWRZ bit (see 6.5.2) is set to:

a) zero, then the device server shall terminate that command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to ZONE IS INACTIVE; and
b) one, then the device server shall:
   A) return logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4; and
   B) not write any data for a read command, if any, that has the FUA bit set to one (see SBC-5).

If a read command or a verify command specifies the reading of one or more LBAs that are in a write pointer zone that has a Zone Condition of OFFLINE, then the device server shall terminate that command with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to ZONE IS OFFLINE.

4.5.3.2 Write pointer zone operations

4.5.3.2.1 Write pointer zone operations overview

The write pointer operations that are applicable to a device server for a zoned block device are summarized in table 14.

Table 14 – Summary of write pointer zone operations

Operation	Description	Reference
Operations defined in this standard for write pointer zones		
close zone operation	Results in the Zone Condition becoming CLOSED	4.5.3.2.3
finish zone operation	Causes subsequent reads that specify logical sectors in the zone that have not been written since the last time the Zone Condition was EMPTY to return the initialization pattern (see 4.5.3.1.4) and results in the Zone Condition becoming FULL	4.5.3.2.4
manage open zone resources operation	Limits the number of open zones, if required, by closing a zone with a Zone Condition of IMPLICITLY OPENED	4.5.3.2.8
open zone operation	Results in the Zone Condition becoming EXPLICITLY OPENED [a]	4.5.3.2.2
reset write pointer operation	Results in the Zone Condition becoming EMPTY	4.5.3.2.5
sequentialize zone operation	Releases zone resources allocated for processing of non-sequential write operations in a zone	4.5.3.2.6
zone activation operation	Detects unmet prerequisites for zone activations and modifies the Zone Conditions of zones being activated and deactivated	4.5.3.2.7
Operations defined in SBC-5 and modified by this standard for write pointer zones		
read operations	Reads logical block data stored in a write pointer zone or initialization pattern (see 4.5.3.1.4)	4.5.3.2.9
write operations	Writes logical block data stored in a write pointer zone and may result in the Zone Condition being IMPLICITLY OPENED [a]	
[a] The zone condition state machine (see 4.5.3.6) requires a manage open zone resources operation to be performed before the Zone Condition is changed to IMPLICITLY OPENED or EXPLICITLY OPENED.		

4.5.3.2.2 Open zone operation

The open zone operation prepares a sequential write required zone (see 4.5.3.4) or sequential write preferred zone (see 4.5.3.3) for writing (e.g., by allocating any open zone resources associated with that zone).

Prior to performing an open zone operation, the zone condition state machine (see 4.5.3.6) requires the device server to:

1) perform a manage open zone resources operation (see 4.5.3.2.8); and
2) not perform that open zone operation if the OZR Available state machine variable (see 4.5.3.6.1) is set to FAILURE.

For a sequential write required zone or sequential write preferred zone, the result of a successful open zone operation is the Zone Condition becoming EXPLICITLY OPENED.

If an open zone operation is performed on:

a) a sequential write required zone or a sequential write preferred zone with a Zone Condition of EXPLICITLY OPENED; or
b) a sequential or before required zone (see 4.5.3.5),

then that operation has no effect and shall not be considered to be an error.

4.5.3.2.3 Close zone operation

A successful close zone operation annuls any preparations made for writing an open sequential write required zone (see 4.5.3.4) or sequential write preferred zone (see 4.5.3.3) (e.g., by deallocating any open zone resources associated with that zone).

For a sequential write required zone or a sequential write preferred zone, performing a close zone operation results in the following becoming non-volatile:

a) any logical block data stored in the specified write pointer zone; and
b) the zone attributes.

For a sequential write required zone or a sequential write preferred zone, the results of a successful close zone operation are:

a) the Zone Condition becoming CLOSED; and
b) all zone resources becoming persistent zone resources.

If a close zone operation is performed on:

a) a sequential write required zone or a sequential write preferred zone with a Zone Condition of CLOSED; or
b) a sequential or before required zone (see 4.5.3.5),

then that operation has no effect and shall not be considered to be an error.

4.5.3.2.4 Finish zone operation

A finish zone operation results in:

a) all logical block data in the zone becoming:
   A) non-volatile; and
   B) available for reading;
   and
b) the Zone Condition becoming FULL.

The zone condition state machine (see 4.5.3.6) requires the specified zone to have a Zone Condition of EXPLICITLY OPENED or IMPLICITLY OPENED before a finish zone operation is performed. If a zone with a Zone Condition of EMPTY or CLOSED is specified for a finish zone operation, prior to processing the finish zone operation, then the zone condition state machine requires that:

a) a manage open zone resources operation (see 4.5.3.2.8) be performed; and
b) the Zone Condition becomes IMPLICITLY OPENED.

The result of a successful finish zone operation is the Zone Condition becoming FULL.

If a finish zone operation is performed on a zone with a Zone Condition of FULL, then that operation has no effect and shall not be considered to be an error.

4.5.3.2.5 Reset write pointer operation

Unless it is performed on a zone associated with a Restrict Write Pointer Reset realm attribute (see 4.4.2) that is true, a reset write pointer operation results in the Zone Condition becoming EMPTY (e.g., the write pointer is set to the lowest LBA in the zone).

If a reset write pointer operation is performed on a zone:

a) with a Zone Condition of EMPTY; or
b) that is associated with a Restrict Write Pointer Reset realm attribute that is true,

then that operation has no effect and shall not be considered to be an error.

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

4.5.3.2.6 Sequentialize zone operation

A successful sequentialize zone operation results in:

a) the release of zone resources allocated for the processing of non-sequential write operations (see 4.5.3.3.2) in the specified write pointer zone; and
b) setting Non-Sequential Write Resources Active (see 4.3.6) to false.

If a sequentialize zone operation is performed on a zone with the Non-Sequential Write Resources Active zone attribute set to false, then that operation has no effect and shall not be considered to be an error.

4.5.3.2.7 Zone activation operation

4.5.3.2.7.1 Zone activation operation overview

As part of processing a ZONE ACTIVATE command (see 5.11) or a ZONE QUERY command (see 5.12), the device server identifies the candidate zones to activate and the candidate zones to deactivate, if any, as defined in 5.11.2.

Of the two sets of candidate zones (i.e., the candidate zones to activate and the candidate zones to deactivate), at least one set contains at least one zone and one set may contain no zones.

The candidate zones to activate and the candidate zones to deactivate are inputs to the zone activation operation as defined in 4.5.3.2.7.

The performing of a zone activation operation shall:

a) prepare the parameter data defined in 5.11.3; and
b) include the performing of:
    1) a verify activations operation (see 4.5.3.2.7.2); and
    2) a change activations operation (see 4.5.3.2.7.3), if
        a) the command that initiated this zone activation operation is a ZONE ACTIVATE command (see 5.11); and
        b) no unmet prerequisites have been detected by the verify activations operation processed in step 1).

4.5.3.2.7.2 Verify activations operation

This subclause (i.e., 4.5.3.2.7.2) describes the process of determining that all candidate zones to activate (see 4.5.3.2.7.1) and candidate zones to deactivate, if any, (see 4.5.3.2.7.1) have appropriate characteristics for activation or deactivation.

Regardless of the value of the ALL bit in the command's CDB, the device server shall:

a) initialize the following bits in the command's parameter data (see 5.11.3) by setting them to zero:
    A) the SECURITY bit;
    B) the NOT EMPTY bit;
    C) the NOT INACT bit;
    D) the MULT DOMN bit;
    E) the RLM RSTCT bit;
    F) the MULT ZTYP bit; and
    G) the RLM ALIGN bit;

and

b) set the SECURITY bit to one, if any of the candidate zones to activate or candidate zones to deactivate, if any, are affected by security restrictions outside the scope of this standard (e.g., specified by a TCG security protocol).

If the ALL bit is set to one, the device server shall:

a) set the NOT EMPTY bit to one, if one or more of the candidate zones to deactivate, if any, have a Zone Condition of CLOSED, EXPLICITLY OPENED, IMPLICITLY OPENED, or FULL; and
b) set the NOT INACT bit to one, if one or more of the candidate zones to activate have a Zone Condition of CLOSED, EXPLICITLY OPENED, IMPLICITLY OPENED, or FULL.

If the ALL bit is set to zero, the device server shall:

a) set the NOT EMPTY bit to one, if one or more of the candidate zones to deactivate, if any, have a Zone Condition that is not EMPTY or INACTIVE (see table 50);
b) set the NOT INACT bit to one, if one or more of the candidate zones to activate have a Zone Condition that is not INACTIVE (see table 50);
c) set the MULT DOMN bit to one, if any of:
   A) the candidate zones to activate include zones in more than one zone domain; or
   B) the candidate zones to deactivate, if any, include zones in more than one zone domain;
d) set the RLM RSTCT bit to one, if the Restrict Zone Activate realm attribute (see 4.4.3) is true for a realm that contains candidate zones to activate, if any, candidate zones to deactivate, if any;
e) set the MULT ZTYP bit to one, if all of the candidate zones to activate, if any, do not have the same Zone Type (see 4.3.2).

If the AAORB bit (see 6.5.2) is set to one and the ALL bit is set to zero, then the device server may set the RLM ALIGN bit to one, if:

a) the candidate zones to activate are not a single zone range;
b) candidate zones to deactivate, if any, are not a single zone range;
c) based on the parameter data returned by a REPORT REALMS command (see 5.6) processed subsequent to the processing of the most recent ZONE ACTIVATE command, FORMAT UNIT command (see SBC-5), or FORMAT WITH PRESET command (see SBC-5) as described in 4.2.4.7:
   A) the candidate zones to activate do not begin and end on realm boundaries; or
   B) candidate zones to deactivate, if any, do not begin and end on realm boundaries.

Unmet prerequisites have been detected if any of the following bits are set to one:

a) the SECURITY bit;
b) the NOT EMPTY bit;
c) the NOT INACT bit;
d) the MULT DOMN bit;
e) the RLM RSTCT bit;
f) the MULT ZTYP bit; or
g) the RLM ALIGN bit.

4.5.3.2.7.3 Change activations operation

This subclause (i.e., 4.5.3.2.7.3) describes the related processes of zone activation and deactivation during the processing of a ZONE ACTIVATE command (see 5.11). The processing in this subclause occurs only after the candidate zones to activate (see 4.5.3.2.7.1) and candidate zones to deactivate, if any, (see 4.5.3.2.7.1) have been determined to have the appropriate characteristics for activation or deactivation (see 4.5.3.2.7.2).

If the candidate zones for activation include more than one zone range, then the device server shall process the zone ranges in increasing LBA order based on the starting LBA for each zone range.

For each zone range in the candidate zones for activation, if the AAORB bit (see 6.5.2) is set to:

a) zero, then the device server shall:

A) activate all of the zones in the zone range to be activated by changing their Zone Condition to EMPTY; and
B) deactivate the zones in the candidate zones for deactivation that correspond to the zone range that has been activated by changing their Zone Condition to INACTIVE;

and

b) one, then for each realm in the zone range to be activated, the device server shall:
A) activate all of the zones in the zone range to be activated in that realm by changing their Zone Condition to EMPTY; and
B) deactivate the zones in the candidate zones for deactivation in that realm by changing their Zone Condition to INACTIVE.

The device server shall set the ACTIVATED bit (see 5.11.3.1) to:

a) one, if all zones were successfully processed; and
b) zero, if no zones were activated or deactivated as a result of performing the change activations operation.

After a power on or hard reset occurs during the performing of a change activations operation the state of the candidate zones to activate and the candidate zones to deactivate depends on the AAORB bit as follows. If the AAORB bit is set to:

a) zero, then:
A) all candidate zones for activation have been activated and all candidate zones for deactivation have been deactivated; or
B) all candidate zones for activation have not been activated and all candidate zones for deactivation have not been deactivated;

and

b) one, then within each realm:
A) all candidate zones for activation have been activated and all candidate zones for deactivation have been deactivated; or
B) all candidate zones for activation have not been activated and all candidate zones for deactivation have not been deactivated.

4.5.3.2.8 Manage open zone resources operation

4.5.3.2.8.1 Manage open zone resources operation overview

The manage open zone resources operation provides a way for the zone condition state machine to ensure that a zone is allowed to become an open zone while maintaining the limits on the number of open zones that the device server indicates in:

a) the OPTIMAL NUMBER OF OPEN SEQUENTIAL WRITE PREFERRED ZONES field (see 6.5.2); and
b) the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field (see 6.5.2).

The manage open zone resources operation is performed only within the zone condition state machine. The manage open zone resources operation sets the OZR Available state machine variable (see 4.5.3.6.1) to:

a) SUCCESS to indicate that sufficient open zone resources are available to allow a Zone Condition to become IMPLICITLY OPENED or EXPLICITLY OPENED; or
b) FAILURE to indicate that insufficient open zone resources are available to allow a Zone Condition to become IMPLICITLY OPENED or EXPLICITLY OPENED.

For a sequential or before required zone (see 4.5.3.5) the device server:

a) shall not close any zones; and
b) shall set the OZR Available state machine variable to SUCCESS.

For write pointer zones other than sequential or before required zones, the manage open zone resources operation:

a) determines whether open zone resources are available to allow a zone to become an open zone; and

b) may close a zone with a Zone Condition of IMPLICITLY OPENED in order to make open zone resources available.

As detailed in 4.5.3.2.8.2 for a sequential write preferred zone (see 4.5.3.3) and in 4.5.3.2.8.3 for a sequential write required zone (see 4.5.3.4), an instance of the manage open zone resources operation sets the OZR Available state machine variable to SUCCESS if:

a) closing a zone is not required in order to limit the number of open zones; or
b) a zone is closed in order to limit the number of open zones,

otherwise, the manage open zone resources operation sets the OZR Available state machine variable to FAILURE.

4.5.3.2.8.2 Select a sequential write preferred zone

For a sequential write preferred zone (see 4.5.3.3), the device server:

a) may choose a sequential write preferred zone with a Zone Condition of IMPLICITLY OPENED and perform a close zone operation (see 4.5.3.2.3) on that zone; and
b) shall set the OZR Available state machine variable to SUCCESS (see 4.5.3.2.8.1),

if the following formula evaluates to true:

$$(k \leq (x + y)) \text{ AND } (y > 0)$$

where:

x is the number of sequential write preferred zones with a Zone Condition of EXPLICITLY OPEN;
y is the number of sequential write preferred zones with a Zone Condition of IMPLICITLY OPEN; and
k is the contents of the OPTIMAL NUMBER OF OPEN SEQUENTIAL WRITE PREFERRED ZONES field (see 6.5.2).

All manage open zone resources operations on sequential write preferred zones set the OZR Available state machine variable to SUCCESS.

4.5.3.2.8.3 Select a sequential write required zone

For a sequential write required zone (see 4.5.3.4), the device server:

a) should not select a zone to be closed; and
b) shall set the OZR Available state machine variable to SUCCESS (see 4.5.3.2.8.1),

if the following formula evaluates to true:

$$z > (x + y)$$

where:

x is the number of sequential write required zones with a Zone Condition of EXPLICITLY OPEN;
y is the number of sequential write required zones with a Zone Condition of IMPLICITLY OPEN; and
z is the contents of the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field (see 6.5.2).

For a sequential write required zone, the device server shall:

a) set the OZR Available state machine variable to FAILURE; and
b) terminate the command that initiated the manage open zone resources operation with CHECK CONDITION status, with sense key set to DATA PROTECT and the additional sense code set to INSUFFICIENT ZONE RESOURCES,

if the following formula evaluates to true:

$$x \geq z$$

where:

x is the number of sequential write required zones with a Zone Condition of EXPLICIT OPEN; and
z is the contents of the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES (see 6.5.2).

For a sequential write required zone, the device server shall:

a) choose a sequential write required zone with a Zone Condition of IMPLICITLY OPENED and perform a close zone operation (see 4.5.3.2.3) on that zone; and
b) set the OZR Available state machine variable to SUCCESS,

if the following formula evaluates to true:

$$(z \leq (x + y)) \text{ AND } (y > 0)$$

where:

x is the number of sequential write required zones with a Zone Condition of EXPLICITLY OPEN;
y is the number of sequential write required zones with a Zone Condition of IMPLICITLY OPEN; and
z is the contents of the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field (see 6.5.2).

4.5.3.2.9 Read operations, verify operations, and write operations

The effects of the write pointer zones zone condition state machine (see 4.5.3.6) on the SBC-5 definitions of the read operation, verify operation, and write operation are reviewed in this subclause (i.e., 4.5.3.2.9).

In zoned block devices, device servers perform read operations, verify operations, and write operations that access write pointer zones (see 4.5.3):

a) as described in the zone condition state machine based on the Zone Condition of the specified write pointer zone;
b) as described in this subclause (i.e., 4.5.3.2.9); and
c) as described in SBC-5.

Before a write operation is allowed to be performed, the zone condition state machine ensures that a write pointer zone has a Zone Condition of EXPLICITLY OPENED or IMPLICITLY OPENED. If prior to performing a write operation the Zone Condition is EMPTY or CLOSED, then:

a) a manage open zone resources operation (see 4.5.3.2.8) is performed; and
b) if the OZR Available state machine variable (see 4.5.3.2.8.1) is set to:
   A) SUCCESS, then the Zone Condition becomes IMPLICITLY OPENED; and
   B) FAILURE, then the write operation is terminated with an error before logical block data is written.

The result of a successful write operation on a write pointer zone with a Zone Condition of EMPTY or CLOSED is a write pointer zone with a Zone Condition of IMPLICITLY OPENED.

The zone condition state machine may or may not require a read operation or a verify operation on a write pointer zone with a Zone Condition of EMPTY or CLOSED to be processed without an error. The result of a successful read operation or a verify operation on a write pointer zone with a Zone Condition of EMPTY or CLOSED does not change the Zone Condition of that write pointer zone.

For write pointer zones with a Zone Condition of INACTIVE or OFFLINE, access pattern requirements (see 4.5.3.1.5, 4.5.3.1.6, 4.5.3.3.2, 4.5.3.3.3, 4.5.3.4.2, 4.5.3.4.3, 4.5.3.5.2, and 4.5.3.5.3) may affect:

a) read operations performed as a result of the device server processing read commands;
b) verify operations performed as a result of the device server processing verify commands; and
c) write operations performed as a result of the device server processing write commands.

For write pointer zones with a Zone Condition of READ ONLY, access pattern requirements (see 4.5.3.1.5, 4.5.3.3.2, 4.5.3.4.2, and 4.5.3.5.2) may affect write operations performed as a result of the device server processing write commands.

4.5.3.3 Sequential write preferred zone model

4.5.3.3.1 Sequential write preferred zone model overview

A sequential write preferred zone is a write pointer zone (see 4.5.3.1) in which the device server allows write commands that specify a starting LBA that is not equal to the associated zone's write pointer. The device server may allocate non-sequential write zone resources for processing these writes (see 4.3.6).

4.5.3.3.2 Write access pattern requirements for sequential write preferred zones

A write command that starts at the write pointer of a sequential write preferred zone and ends at the highest LBA of that sequential write preferred zone results in the Zone Condition changing to FULL.

Figure 12 shows an example of a write command that starts at the write pointer of a sequential write preferred zone.

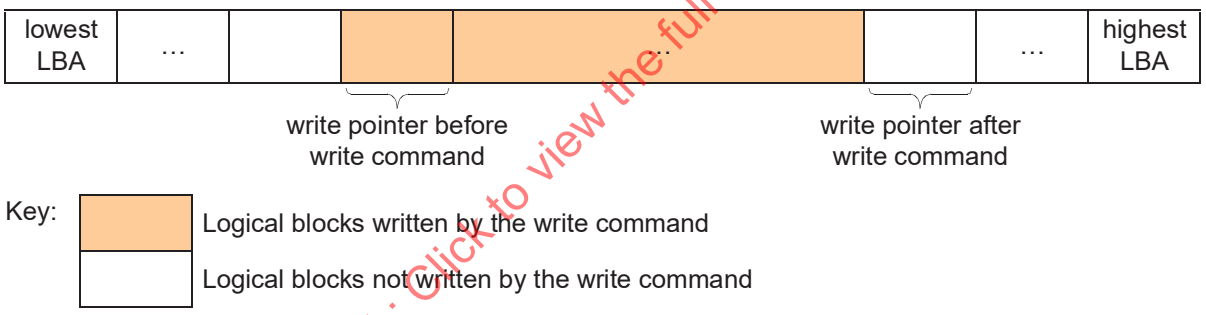

Figure 12 – Example write command that starts at the write pointer

Figure 13 shows examples of write commands that do not start at the write pointer and, as a result, cause at least one non-sequential write operation to be performed.

A non-sequential write operation shall result in:

a) the associated zone's write pointer being set to an LBA that is:
   A) greater than or equal to the value of that write pointer before that operation; and
   B) less than or equal to the minimum of:
      a) one plus the highest LBA written in the zone since the last time the Zone Condition was EMPTY; and
      b) the highest LBA in the zone;

   or
b) the Zone Condition becoming FULL (e.g., the write pointer becoming invalid).

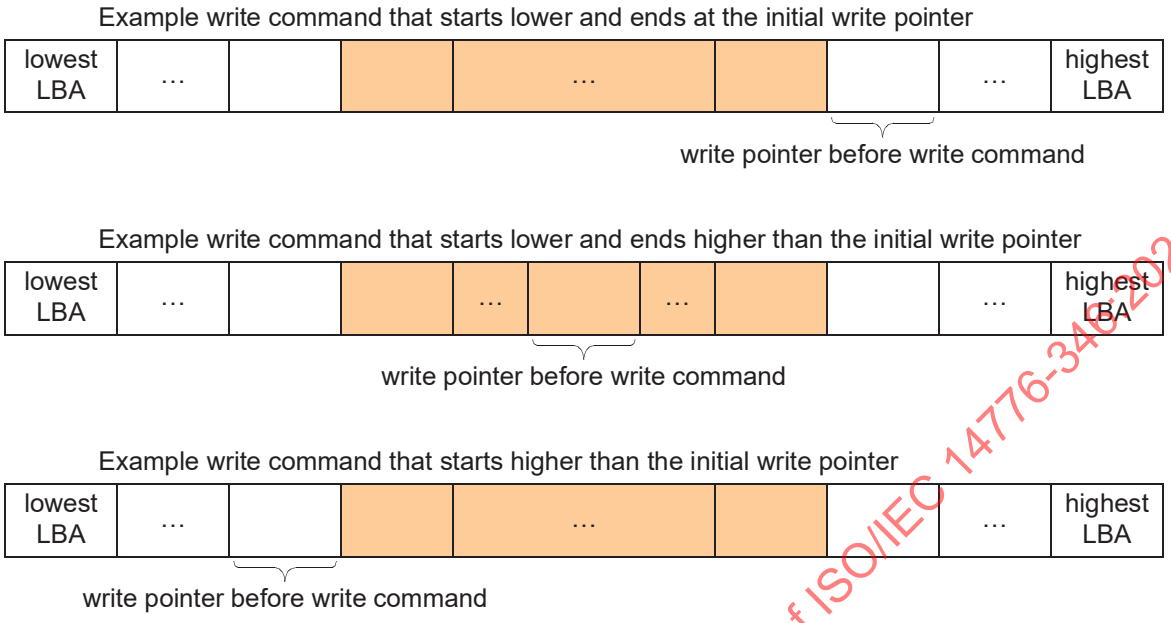

Figure 13 – Examples of write commands that do not start at the write pointer

If the device server processes a non-sequential write operation to a sequential write preferred zone, then the device server shall:

a) set the associated zone's write pointer to an LBA that is:
   A) greater than or equal to the value of that write pointer before that command was processed; and
   B) less than or equal to the minimum of:
      a) one plus the highest LBA written in the zone since the last time the Zone Condition was EMPTY; and
      b) the highest LBA in the zone;

   or
b) cause the Zone Condition to become FULL.

If a non-sequential write command completes without error and causes the Zone Condition to become FULL, then the device server shall complete that command or a subsequent command accessing that zone with GOOD status with the sense key set to COMPLETED and the additional sense code set to ZONE TRANSITION TO FULL.

The device server may incur delays during the processing of subsequent commands if the device server processes a command that results in:

a) the number of open sequential write preferred zones exceeding the value in the OPTIMAL NUMBER OF OPEN SEQUENTIAL WRITE PREFERRED ZONES field (see 6.5.2); or
b) the number of non-sequentially written sequential write preferred zones exceeding the value in the OPTIMAL NUMBER OF NON-SEQUENTIALLY WRITTEN SEQUENTIAL WRITE PREFERRED ZONES field (see 6.5.2).

Regardless of the starting LBA in a write command, the resulting value of the write pointer is outside the scope of this standard if:

a) at least one logical block has been written;

b) not all the logical blocks specified by the command have been written; and
c) write caching:
   A) is not enabled and the write command is terminated with an error; or
   B) is enabled and an error occurs after the command completes with success.

Write command requirements that affect all types of write pointer zones are described in 4.5.3.1.5.

4.5.3.3.3 Read access pattern requirements for sequential write preferred zones

A read command that accesses an LBA that has been written since the most recent time the Zone Condition was EMPTY for that sequential write preferred zone shall return the most recent written logical block data. A read command that accesses an LBA that has not been written since the most recent time the Zone Condition was EMPTY for that sequential write preferred zone shall:

a) return logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4; and
b) not write any data for a read command that has the FUA bit set to one (see SBC-5).

A verify command that accesses an LBA that has been written since the most recent time the Zone Condition was EMPTY for that sequential write preferred zone shall use the most recent written data for that verify operation. A verify command that accesses an LBA that has not been written since the most recent time the Zone Condition was EMPTY for that sequential write preferred zone shall use logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4.

Read command requirements and verify command requirements that affect all types of write pointer zones are described in 4.5.3.1.6.

4.5.3.4 Sequential write required zone model

4.5.3.4.1 Sequential write required zone model overview

A sequential write required zone is a write pointer zone (see 4.5.3.1) in which the device server requires that write commands specify a starting LBA that is equal to that zone's write pointer.

4.5.3.4.2 Write access pattern requirements for sequential write required zones

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 sequential write required zone, then the Zone Condition becomes FULL and the write pointer becomes invalid.

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. The application client should use the REPORT ZONES command (see 5.8) to determine the current write pointer and other zone attributes of the sequential write required zone.

If the value in the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field (see 6.5.2) is non-zero and the number of zones with a Zone Condition of EXPLICITLY OPENED is equal to the value in the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field, then a command that writes to or attempts to open a sequential write required zone with a zone condition of EMPTY or CLOSED is terminated with CHECK CONDITION status with sense key set to DATA PROTECT and the additional sense code set to INSUFFICIENT ZONE RESOURCES (see 4.5.3.2.8).

An entire medium write same command is a WRITE SAME (16) command with:

a) the LOGICAL BLOCK ADDRESS field set to zero; and
b) the NUMBER OF LOGICAL BLOCKS field set to:
   A) zero (i.e., all logical blocks, ending with the last logical block on the medium); or

B) the LBA of the last logical block on the logical unit plus one.

If the WSNZ bit is set to one in the Block Limits VPD page (see SBC-5) or the device server does not support the NUMBER OF LOGICAL BLOCKS field set to the LBA of the last logical block on the logical unit plus one, then entire medium write same commands are not supported.

If the device server processes a write command with the starting LBA in a sequential write required zone with the Zone Condition of FULL, then the device server terminates that command with CHECK CONDITION status, the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The device server terminates with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to UNALIGNED WRITE COMMAND a write command, other than an entire medium write same command, that specifies:

a) the starting LBA in a sequential write required zone set to a value that is not equal to the write pointer for that sequential write required zone; or
b) an ending LBA that is not equal to the last logical block within a physical block (see SBC-5).

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

Write command requirements that affect all types of write pointer zones are described in 4.5.3.1.5.

If the device server terminates a command with the additional sense code set to UNALIGNED WRITE COMMAND or WRITE BOUNDARY VIOLATION, then the device server shall return the valid 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-6).

In a sequential write required zone, if the device server is not able to write all of the logical blocks specified by a write command before encountering an error that prevents further writing to the media, then the resulting value of the write pointer is outside the scope of this standard. The host should use the REPORT ZONES command (see 5.8) to determine the current write pointer and other characteristics of the sequential write required zone.

If the device server processes a WRITE LONG command with the WR_UNCOR bit set to one and the starting LBA equal to the write pointer of a sequential write required zone, then the device server shall:

a) mark all of the logical blocks in that physical block as uncorrectable;
b) modify the write pointer to be one greater than the LBA of the highest numbered logical block that was marked as uncorrectable, if the highest numbered logical block that was marked as uncorrectable is not the highest numbered logical block in that zone; and
c) change the Zone Condition to FULL, if the highest numbered logical block that was marked as uncorrectable is the highest numbered logical block in that zone.

EXAMPLE - For a sequential write required zone in a zoned block device with 8 logical blocks per physical block and the first LBA in a physical block is LBA 8. A WRITE LONG command with LBA 8 results in LBAs 8 through 15 being marked as uncorrectable and the write pointer being set to LBA 16.

4.5.3.4.3 Read access pattern requirements for sequential write required zones

If the URSWRZ bit (see 6.5.2) is set to one and the device server processes a read command or a verify command:

a) with the starting LBA:
   A) less than the write pointer and the ending LBA greater than or equal to the write pointer in the same sequential write required zone; or
   B) greater than or equal to the write pointer and the ending LBA in the same sequential write required zone;
   or

b) with:
   A) the ending LBA in a different sequential write required zone than the starting LBA; and
   B) each LBA specified by the command in a sequential write required zone,

then, for any LBA that:

a) contains logical block data written since the most recent time the Zone Condition was EMPTY for the sequential write required zone being read or verified, the device server shall process the command using the logical block data; or
b) does not contain logical block data written since the most recent time the Zone Condition was EMPTY for the sequential write required zone being read or verified, the device server shall:
   A) for an LBA that is being verified, use the logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4; and
   B) for an LBA that is being read:
      a) return logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4; and
      b) not write any data for a read command that has the FUA bit set to one (see SBC-5).

Read command requirements and verify command requirements that affect all types of write pointer zones are described in 4.5.3.1.6.

If the device server terminates a command with the additional sense code set to ATTEMPT TO READ INVALID DATA or READ BOUNDARY VIOLATION, then the device server shall return the valid write pointer, if any, associated with the sequential write required zone, if any, specified by the LOGICAL BLOCK ADDRESS field of that command in the INFORMATION field of the sense data (see SPC-6).

4.5.3.4.4 Opening Sequential Write Required zones

If the number of sequential write required zones with Zone Condition EXPLICITLY OPENED is equal to the value in the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field (see 6.5.2), then any command (e.g., a write command) that attempts to cause a sequential write required zone that is not an open zone to become an open zone is terminated with an error as described in the zone condition state machine (see 4.5.3.6).

4.5.3.5 Sequential or before required zone model

4.5.3.5.1 Sequential or before required zone model overview

A sequential or before required zone is a write pointer zone in which the device server allows write commands specify a starting LBA that is equal to or less than that zone's write pointer.

4.5.3.5.2 Write access pattern requirements for sequential or before required zones

If the device server processes a write command without error and the ending LBA is greater than or equal to the write pointer in the last sequential write required zone written, then the write pointer for that sequential or before required zone is set to the ending LBA plus one (i.e., the write pointer minus one indicates highest LBA for which logical block data is available to be read).

If the ending LBA of a write command that completed without error is equal to the highest LBA of a sequential or before required zone, then the Zone Condition becomes FULL and the write pointer becomes invalid.

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. The application client should use the REPORT ZONES command (see 5.8) to determine the current write pointer and other zone attributes of the sequential or before required zone.

The device server terminate with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to UNALIGNED WRITE COMMAND a write command that:

a) specifies a starting LBA in a sequential or before required zone set to a value that is greater than the write pointer for that sequential or before required zone; or
b) attempts to cause the sequential or before required zone specified by the ending LBA to have a valid write pointer that is not equal to the first logical sector within a physical sector (see SBC-5).

Write command requirements that affect all types of write pointer zones are described in 4.5.3.1.5.

If the device server terminates a command with the additional sense code set to UNALIGNED WRITE COMMAND or WRITE BOUNDARY VIOLATION, then the device server shall return the valid write pointer, if any, associated with the sequential or before required zone specified by the LOGICAL BLOCK ADDRESS field of that command in the INFORMATION field of the sense data (see SPC-6).

In a sequential or before required zone, if the device server is not able to write all of the logical blocks specified by a write command before encountering an error that prevents further writing to the media and the ending LBA specified by the write command is:

a) less than the write pointer, then the write pointer shall not be changed; and
b) greater than or equal to the write pointer, then the resulting value of the write pointer is outside the scope of this standard.

If the device server processes a WRITE LONG command with the WR_UNCOR bit set to one and the starting LBA less than or equal to the write pointer of a sequential or before required zone, then the device server shall:

a) mark all of the logical blocks in that physical block as uncorrectable;
b) modify the write pointer to be one greater than the LBA of the highest numbered logical block that was marked as uncorrectable, if the highest numbered logical block that was marked as uncorrectable is not the highest numbered logical block in that zone; and
c) change the Zone Condition to FULL, if the highest numbered logical block that was marked as uncorrectable is the highest numbered logical block in that zone.

EXAMPLE - For a sequential or before required zone in a zoned block device with 8 logical blocks per physical block and the first LBA in a physical block is LBA 8. A WRITE LONG command with LBA 8 results in LBAs 8 through 15 being marked as uncorrectable and the write pointer being set to LBA 16.

4.5.3.5.3 Read access pattern requirements for sequential or before required zones

The device server shall terminate a read command or a verify 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:

a) the URSWRZ bit (see 6.5.2) is set to zero; and
b) that command specifies an LBA:
   A) in a sequential or before required zone that has a Zone Condition of EMPTY; or
   B) that is greater than or equal to the write pointer in a sequential or before required zone that does not have a Zone Condition of FULL.

If the URSWRZ bit is set to one and the device server processes a read command or a verify command with no other errors (e.g., the errors described in 4.5.3.1.6), then, for any LBA that:

a) contains logical block data written since the most recent time the Zone Condition was EMPTY for the sequential write required zone being read or verified, the device server shall process the command using the logical block data; or
b) does not contain logical block data written since the most recent time the Zone Condition was EMPTY for the sequential write required zone being read or verified, the device server shall:
   A) for an LBA that is being verified, use the logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4; and
   B) for an LBA that is being read:

a) return logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4; and
b) not write any data for a read command that has the FUA bit (see SBC-5) set to one.

Read command requirements and verify command requirements that affect all types of write pointer zones are described in 4.5.3.1.6.

If the device server terminates a command with the additional sense code set to ATTEMPT TO READ INVALID DATA or READ BOUNDARY VIOLATION, then the device server shall return the valid write pointer, if any, associated with the sequential or before required zone, if any, specified by the LOGICAL BLOCK ADDRESS field of that command in the INFORMATION field of the sense data (see SPC-6).

4.5.3.6 Zone condition state machine

4.5.3.6.1 Zone condition state machine overview

There is one zone condition state machine for each write pointer zone. The zone condition state machine (see figure 14) controls the operation of each write pointer zone. The zone condition state machine consists of the following states:

a) ZC1:Empty (see 4.5.3.6.2);
b) ZC2:Implicit_Open (see 4.5.3.6.3);
c) ZC3:Explicit_Open (see 4.5.3.6.4);
d) ZC4:Closed (see 4.5.3.6.5);
e) ZC5:Full (see 4.5.3.6.6);
f) ZC6:Read_Only (see 4.5.3.6.7);
g) ZC7:Offline (see 4.5.3.6.8); and
h) ZC8:Inactive (see 4.5.3.6.9).

Each zone condition state machine is an integral part of the device server that participates in the processing of specified commands and in the performing of specified operations. The processing of a single command (e.g., a FINISH ZONE command (see 5.3)) may include zero or more transitions in the zone condition state machine. Information about device server progress in the processing of such commands affects transitions in the zone condition state machine.

Unless otherwise specified in the zone condition state machine, the device server shall maintain zone condition state after any condition changes or SCSI events (see SAM-6) except for a power on.

The initial state for a write pointer zone after power on is:

a) the ZC1:Empty state if:
   A) the zone type is sequential or before required and the most recent Zone Condition was EMPTY; or
   B) the zone type is not sequential or before required and:
      a) the write pointer is valid and indicates the lowest LBA in the zone; and
      b) Non-Sequential Write Resources Active (see 4.3.6) is false;
b) the ZC2:Implicit_Open state if the zone type is sequential or before required (see 4.5.3.5) and the most recent Zone Condition was IMPLICITLY OPENED;
c) the ZC4:Closed state, if the zone type is not sequential or before required and:
   A) the write pointer is valid and indicates an LBA that is not the lowest LBA in the zone; or
   B) Non-Sequential Write Resources Active is true;
d) the ZC5:Full state, if the most recent Zone Condition was FULL;
e) the ZC6:Read_Only state, if the most recent Zone Condition was READ ONLY;
f) the ZC7:Offline state, if the most recent Zone Condition was OFFLINE; and
g) the ZC8:Inactive state, if the most recent Zone Condition was INACTIVE.

Upon completion of a sanitize operation the Zone Condition (i.e., zone state) of each write pointer zone is described in 4.11.

This state machine shall maintain the OZR Available state machine variable to indicate the success or failure of the most recent manage open zone resources operation (see 4.5.3.2.8).

The characteristics associated with the state of a zone are summarized in table 15.

Table 15 – Characteristics associated with zone state

State	Zone characteristics			
	Write pointer valid [a]	Open zone resources in use [b]	Accessible for	
			Reads	Writes
ZC1:Empty state	Yes	No	see [c]	see [d]
ZC2:Implicit_Open state	Yes	Yes	Yes	Yes
ZC3:Explicit_Open state	Yes	Yes	Yes	Yes
ZC4:Closed state	Yes	No	Yes	see [d]
ZC5:Full state	No	No	Yes	see [e]
ZC6:Read_Only state	No	No	Yes	No
ZC7:Offline state	No	No	No	No
ZC8:Inactive state	No	No	see [c]	No

[a] A valid write pointer (i.e., Yes) indicates a specific LBA in the zone as described in 4.5.3.1. An invalid write pointer (i.e., No) provides no information.

[b] See 4.5.3.1.3.

[c] This zone is accessible for reads if:

a) the zone type is sequential write preferred (see 4.5.3.3.3); or

b) the URSWRZ bit (see 6.5.2) is set to one and the zone type is:

A) sequential write required (see 4.5.3.4.3); or

B) sequential or before required (see 4.5.3.5.3).

[d] This zone is accessible for writes if the manage open zone resources operation (see 4.5.3.2.8) is able to complete with success and the zone transitions to the ZC2:Implicit_Open state.

[e] This zone is accessible for writes if the zone type is sequential write preferred (see 4.5.3.3.2) or sequential or before required (see 4.5.3.5.2).

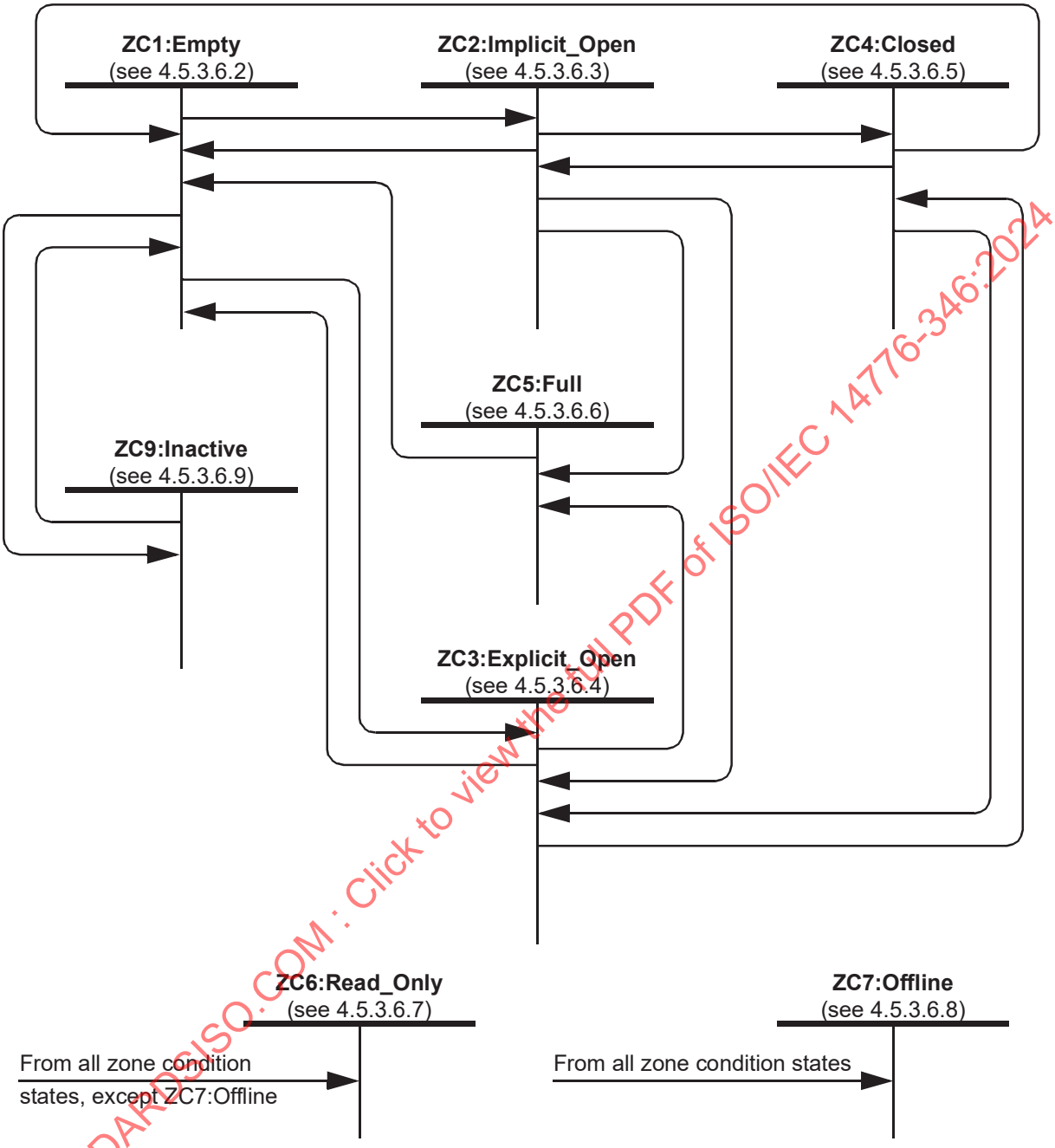

Figure 14 – Zone condition state machine

4.5.3.6.2 ZC1:Empty state

4.5.3.6.2.1 ZC1:Empty state overview

While in this state:

a) Zone Condition shall be set to EMPTY;
b) the write pointer is valid and indicates the lowest LBA in the zone;
c) all zone resources (e.g., the write pointer) shall be persistent zone resources (i.e., open zone resources shall not be used);
d) RWP Recommended is set as defined in 4.3.5;
e) Non-Sequential Write Resources Active is set as defined in 4.3.6;
f) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
g) if the device server begins to perform:
   A) a write operation;
   B) a finish zone operation; or
   C) an open zone operation (i.e., as part of processing an OPEN ZONE command),
   then, the device server shall:
   1) perform a manage open zone resources operation (see 4.5.3.2.8), including the setting of the OZR Available state machine variable (see 4.5.3.6.1); and
   2) if the OZR Available state machine variable is set to FAILURE, then the device server shall terminate the command that initiated the manage open zone resources operation with CHECK CONDITION status, with sense key set to DATA PROTECT and the additional sense code set to INSUFFICIENT ZONE RESOURCES;
   and
h) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in:
   A) 4.5.3.3.3 for sequential write preferred zones;
   B) 4.5.3.4.3 for sequential write required zones; and
   C) 4.5.3.5.3 for sequential or before required zones.

4.5.3.6.2.2 Transition ZC1:Empty to ZC2:Implicit_Open

This transition shall occur if:

1) as described in 4.5.3.6.2.1, the device server begins to perform:
   A) a write operation; or
   B) a finish zone operation;
   and
2) the OZR Available state machine variable (see 4.5.3.6.1) is set to SUCCESS.

4.5.3.6.2.3 Transition ZC1:Empty to ZC3:Explicit_Open

This transition shall occur if:

1) as described in 4.5.3.6.2.1, the device server performs an open zone operation (i.e., as part of processing an OPEN ZONE command); and
2) the OZR Available state machine variable (see 4.5.3.6.1) is set to SUCCESS.

4.5.3.6.2.4 Transition ZC1:Empty to ZC6:Read_Only

This transition may occur for reasons outside the scope of this standard.

4.5.3.6.2.5 Transition ZC1:Empty to ZC7:Offline

This transition may occur:

a) as a result of media failure (see 4.6.4);

b) as the result of processing a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); or
c) for reasons outside the scope of this standard.

4.5.3.6.2.6 Transition ZC1:Empty to ZC8:Inactive

This transition shall occur upon successful completion of a change activations operation (see 4.5.3.2.7.3) that deactivates this zone.

4.5.3.6.3 ZC2:Implicit_Open state

4.5.3.6.3.1 ZC2:Implicit_Open state overview

While in this state:

a) Zone Condition shall be set to IMPLICITLY OPENED;
b) the write pointer (see 4.5.3.1) is valid;
c) all zone resources (e.g., the write pointer) shall be open zone resources, persistent zone resources, or both (i.e., this zone uses the open zone resources allocated to it);
d) RWP Recommended zone shall be maintained as defined in 4.3.5;
e) Non-Sequential Write Resources Active shall be maintained as defined in 4.3.6;
f) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
g) if the device server performs a write operation, then the device server shall perform the write operation and process the write command that resulted in the write operation as described in:
   A) 4.5.3.3.2 for sequential write preferred zones;
   B) 4.5.3.4.2 for sequential write required zones; and
   C) 4.5.3.5.2 for sequential or before required zones;
   and
h) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in:
   A) 4.5.3.3.3 for sequential write preferred zones;
   B) 4.5.3.4.3 for sequential write required zones; and
   C) 4.5.3.5.3 for sequential or before required zones.

4.5.3.6.3.2 Transition ZC2:Implicit_Open to ZC1:Empty

This transition shall occur:

a) upon successful completion of a reset write pointer operation; or
b) if the write pointer is at the lowest LBA of the zone and Non-Sequential Write Resources Active (see 4.3.6) is false, then upon successful completion of:
   A) a close zone operation for this zone; or
   B) a manage open zone resources operation for a different zone that selects this zone to be closed.

4.5.3.6.3.3 Transition ZC2:Implicit_Open to ZC3:Explicit_Open

This transition shall occur upon successful completion of an open zone operation (i.e., as part of processing an OPEN ZONE command).

4.5.3.6.3.4 Transition ZC2:Implicit_Open to ZC4:Closed

If the write pointer is not at the lowest LBA of the zone or Non-Sequential Write Resources Active (see 4.3.6) is true, then this transition shall occur upon successful completion of:

a) a close zone operation for this zone; or
b) a manage open zone resources operation for a different zone that selects this zone to be closed.

4.5.3.6.3.5 Transition ZC2:Implicit_Open to ZC5:Full

For a sequential write required zone, this transition shall occur upon successful completion of:

a) a finish zone operation in this zone; or
b) a write operation that writes from the write pointer to the highest LBA in the zone.

For a sequential write preferred zone, this transition:

a) may occur upon successful completion of a non-sequential write operation (see 4.5.3.3.2);
b) shall occur upon successful completion of a write operation that writes from the write pointer to the highest LBA in this zone; and
c) shall occur upon a successful completion of a finish zone operation in this zone.

For a sequential or before required zone, this transition shall occur upon successful completion of a write operation with a starting LBA less than or equal to the write pointer that writes the highest LBA in this zone.

4.5.3.6.3.6 Transition ZC2:Implicit_Open to ZC6:Read_Only

This transition may occur for reasons outside the scope of this standard.

4.5.3.6.3.7 Transition ZC2:Implicit_Open to ZC7:Offline

This transition may occur:

a) as a result of media failure (see 4.6.4);
b) as the result of processing a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); or
c) for reasons outside the scope of this standard.

4.5.3.6.4 ZC3:Explicit_Open state

4.5.3.6.4.1 ZC3:Explicit_Open state overview

While in this state:

a) Zone Condition shall be set to EXPLICITLY OPENED;
b) the write pointer (see 4.5.3.1) is valid;
c) all zone resources (e.g., the write pointer) shall be open zone resources, persistent zone resources, or both (i.e., this zone uses the open zone resources allocated to it);
d) RWP Recommended shall be maintained as defined in 4.3.5;
e) Non-Sequential Write Resources Active shall be maintained as defined in 4.3.6;
f) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
g) if the device server performs a write operation, then the device server shall perform the write operation and process the write command that resulted in the write operation as described in:
   A) 4.5.3.3.2 for sequential write preferred zones; and
   B) 4.5.3.4.2 for sequential write required zones;
   and
h) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in:
   A) 4.5.3.3.3 for sequential write preferred zones; and
   B) 4.5.3.4.3 for sequential write required zones.

4.5.3.6.4.2 Transition ZC3:Explicit_Open to ZC1:Empty

This transition shall occur upon successful completion of:

a) a reset write pointer operation; or
b) a close zone operation, if the write pointer is at the lowest LBA of the zone and Non-Sequential Write Resources Active (see 4.3.6) is false.

4.5.3.6.4.3 Transition ZC3:Explicit_Open to ZC4:Closed

This transition shall occur upon successful completion of a close zone operation, if the write pointer is not at the lowest LBA of the zone or Non-Sequential Write Resources Active (see 4.3.6) is true.

4.5.3.6.4.4 Transition ZC3:Explicit_Open to ZC5:Full

For a sequential write required zone, this transition shall occur upon successful completion of:

a) a finish zone operation in this zone; or
b) a write operation that writes from the write pointer to the highest LBA in the zone.

For a sequential write preferred zone, this transition:

a) may occur upon successful completion of a non-sequential write operation (see 4.5.3.3.2);
b) shall occur upon successful completion of a write operation that writes from the write pointer to the highest LBA in this zone; and
c) shall occur upon a successful completion of a finish zone operation in this zone.

4.5.3.6.4.5 Transition ZC3:Explicit_Open to ZC6:Read_Only

This transition may occur for reasons outside the scope of this standard.

4.5.3.6.4.6 Transition ZC3:Explicit_Open to ZC7:Offline

This transition may occur:

a) as a result of media failure (see 4.6.4);
b) as the result of processing a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); or
c) for reasons outside the scope of this standard.

4.5.3.6.5 ZC4:Closed state

4.5.3.6.5.1 ZC4:Closed state overview

While in this state:

a) Zone Condition shall be set to CLOSED;
b) the write pointer (see 4.5.3.1) is valid;
c) all zone resources (e.g., the write pointer) shall be persistent zone resources (i.e., open zone resources shall not be used);
d) all logical block data shall be stored on the medium or in non-volatile cache;
e) RWP Recommended shall be maintained as defined in 4.3.5;
f) Non-Sequential Write Resources Active shall be maintained as defined in 4.3.6;
g) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
h) if the device server begins to perform:
   A) a write operation;
   B) a finish zone operation; or
   C) an open zone operation (i.e., as part of processing an OPEN ZONE command),
   then, the device server shall:
   1) perform a manage open zone resources operation (see 4.5.3.2.8), including the setting of the OZR Available state machine variable (see 4.5.3.6.1); and
   2) if the OZR Available state machine variable is set to FAILURE, then the device server shall terminate the command that initiated the manage open zone resources operation with CHECK CONDITION status, with sense key set to DATA PROTECT and the additional sense code set to INSUFFICIENT ZONE RESOURCES;

   and
i) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in:

A) 4.5.3.3.3 for sequential write preferred zones; and
B) 4.5.3.4.3 for sequential write required zones.

4.5.3.6.5.2 Transition ZC4:Closed to ZC1:Empty

This transition shall occur upon successful completion of a reset write pointer operation.

4.5.3.6.5.3 Transition ZC4:Closed to ZC2:Implicit_Open

This transition shall occur if:

1) as described in 4.5.3.6.2.1, the device server begins to perform:
   A) a write operation; or
   B) a finish zone operation;
   and
2) the OZR Available state machine variable (see 4.5.3.6.1) is set to SUCCESS.

4.5.3.6.5.4 Transition ZC4:Closed to ZC3:Explicit_Open

This transition shall occur if:

1) as described in 4.5.3.6.2.1, the device server performs an open zone operation (i.e., as part of processing an OPEN ZONE command); and
2) the OZR Available state machine variable (see 4.5.3.6.1) is set to SUCCESS.

4.5.3.6.5.5 Transition ZC4:Closed to ZC6:Read_Only

This transition may occur for reasons outside the scope of this standard.

4.5.3.6.5.6 Transition ZC4:Closed to ZC7:Offline

This transition may occur:

a) as a result of media failure (see 4.6.4);
b) as the result of processing a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); or
c) for reasons outside the scope of this standard.

4.5.3.6.6 ZC5:Full state

4.5.3.6.6.1 ZC5:Full state overview

While in this state:

a) Zone Condition shall be set to FULL;
b) the write pointer is invalid;
c) all LBAs in the zone are able to return logical block data in response to a read command;
d) all zone resources (e.g., the write pointer) shall be persistent zone resources (i.e., open zone resources shall not be used);
e) all logical block data shall be stored on the medium or in non-volatile cache;
f) RWP Recommended shall be maintained as defined in 4.3.5;
g) Non-Sequential Write Resources Active shall be maintained as defined in 4.3.6;
h) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
i) if the device server performs a write operation, then the device server shall perform the write operation and process the write command that resulted in the write operation as described in:
   A) 4.5.3.3.2 for sequential write preferred zones;
   B) 4.5.3.4.2 for sequential write required zones; and
   C) 4.5.3.5.2 for sequential or before required zones;
   and

j) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in:
   A) 4.5.3.3.3 for sequential write preferred zones;
   B) 4.5.3.4.3 for sequential write required zones; and
   C) 4.5.3.5.3 for sequential or before required zones.

4.5.3.6.6.2 Transition ZC5:Full to ZC1:Empty

This transition shall occur upon successful completion of a reset write pointer operation.

4.5.3.6.6.3 Transition ZC5:Full to ZC6:Read_Only

This transition may occur for reasons outside the scope of this standard.

4.5.3.6.6.4 Transition ZC5:Full to ZC7:Offline

This transition may occur:

a) as a result of media failure (see 4.6.4);
b) as the result of processing a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); or
c) for reasons outside the scope of this standard.

4.5.3.6.7 ZC6:Read_Only state

4.5.3.6.7.1 ZC6:Read_Only state overview

While in this state:

a) Zone Condition shall be set to READ ONLY;
b) the write pointer is invalid;
c) all zone resources (e.g., the write pointer) shall be persistent zone resources (i.e., open zone resources shall not be used);
d) RWP Recommended is set as defined in 4.3.5;
e) Non-Sequential Write Resources Active is set as defined in 4.3.6;
f) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
g) if the device server performs a write operation or a finish zone operation, then the device server shall terminate the command that resulted in the operation with CHECK CONDITION status, with the sense key set to DATA PROTECT and additional sense code set to ZONE IS READ ONLY; and
h) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in:
   A) 4.5.3.3.3 for sequential write preferred zones;
   B) 4.5.3.4.3 for sequential write required zones; and
   C) 4.5.3.5.3 for sequential or before required zones.

With the exception of transitions to the ZC7:Offline state, transitions out of this state are outside the scope of this standard.

4.5.3.6.7.2 Transition ZC6:Read_Only to ZC7:Offline

This transition may occur:

a) as a result of media failure (see 4.6.4);
b) as the result of performing a sanitize operation (see 4.11);
c) as the result of processing a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5); or
d) for reasons outside the scope of this standard.

4.5.3.6.8 ZC7:Offline state

While in this state:

a) Zone Condition shall be set to OFFLINE;
b) the write pointer is invalid;
c) all zone resources (e.g., the write pointer) shall be persistent zone resources (i.e., open zone resources shall not be used);
d) RWP Recommended is set as defined in 4.3.5;
e) Non-Sequential Write Resources Active is set as defined in 4.3.6; and
f) if the device server performs a read operation, a write operation, or a finish zone operation, then the device server shall terminate the command that resulted in the operation with CHECK CONDITION status, with the sense key set to DATA PROTECT, and additional sense code set to ZONE IS OFFLINE.

All transitions out of this state are outside the scope of this standard.

4.5.3.6.9 ZC8:Inactive state

4.5.3.6.9.1 ZC8:Inactive state overview

While in this state:

a) Zone Condition shall be set to INACTIVE;
b) the write pointer is invalid;
c) all zone resources (e.g., the write pointer) shall be persistent zone resources (i.e., open zone resources shall not be used);
d) RWP Recommended is set as defined in 4.3.5;
e) Non-Sequential Write Resources Active is set as defined in 4.3.6;
f) Predicted Unrecovered Errors Present is set as defined in 4.3.7;
g) if the device server performs a write operation or a finish zone operation, then the device server shall terminate the command that resulted in the operation with CHECK CONDITION status, with the sense key set to DATA PROTECT, and additional sense code set to ZONE IS INACTIVE; and
h) if the device server performs a read operation, then the device server shall perform the read operation and process the read command that resulted in the read operation as described in 4.5.3.1.6.

4.5.3.6.9.2 Transition ZC8:Inactive to ZC1:Empty state

This transition shall occur upon successful completion of a change activations operation (see 4.5.3.2.7.3) that activates this zone.

4.5.4 Gap zone model

For gap zones, the ZONE CONDITION field (see 5.8.2) shall be set to NOT WRITE POINTER (see table 44).

Gap zones:

a) shall not be reported by:
   A) the REPORT ZONE DOMAINS command (see 5.7); or
   B) the REPORT REALMS command (see 5.6);
   and
b) shall be reported by the REPORT ZONES command (see 5.8).

If a write command specifies the writing of one or more LBAs that are in a gap zone, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to ATTEMPT TO ACCESS GAP ZONE and the INFORMATION field (see SPC-6) set to the lesser of:

a) the starting LBA of the write command: or
b) the zone ID of the first accessed gap zone.

If a read command specifies the reading of one or more LBAs that are in a gap zone and the URSWRZ bit (see 6.5.2) is set to:

a) zero, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, the additional sense code set to ATTEMPT TO ACCESS GAP ZONE and the INFORMATION field set to the lesser of:
   A) the starting LBA of the read command: or
   B) the zone ID of the first accessed gap zone;
   and
b) one, then the device shall process all read command accesses to LBAs that are in a gap zone using the initialization pattern as described in 4.5.3.1.4.

4.6 Zoned block device extensions to block device model

4.6.1 Overview

Zoned block devices extend some capabilities defined in SBC-5 as described in 4.6.

4.6.2 Zoned block device internal resource management

A zoned block device requires internal resources (e.g., persistent zone 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).

A RESET bit set to one in a zone descriptor (see 5.8.2) indicates that an insufficient resources condition has been detected for that zone.

An application client that detects the RESET bit being set to one may respond by sending a RESET WRITE POINTER command (see 5.9) that specifies that zone. Before sending the RESET WRITE POINTER command, the application client may copy the logical block data in the affected zone to another location.

4.6.3 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-6) and:

a) there are partially completed write operations;
b) there is logical block data in volatile write cache; or
c) zone attributes for completed write operations are not stored in persistent zone resources,

then an unexpected power removal condition has occurred.

An unexpected power removal condition may result in zone attributes for partially completed write commands to be updated to reflect the partial logical block data written to the medium (e.g., a write pointer may indicate the LBA plus one of the last logical block that the device server attempted to write before the unexpected power removal).

If:

1) write caching is disabled (see SBC-5);
2) a write command completes without error; and
3) an unexpected power removal occurs,

then, after power is restored, a read command that completes without error for an LBA included in that write command shall return the logical block data that was written by that write command before the unexpected power removal.

If:

1) a write command with the FUA bit set to one (see SBC-5) completes without error; and
2) an unexpected power removal occurs,

then, after power is restored, a read command that completes without error for an LBA included in that write command shall return the logical block data that was written by that write command before the unexpected power removal.

If:

1) a write command completes without error;
2) a SYNCHRONIZE CACHE command (see SBC-5) completes without error; and
3) an unexpected power removal occurs,

then, after power is restored, a read command that completes without error for an LBA included in that write command shall return the logical block data that was written by that write command before the unexpected power removal.

If:

1) a verify command (see SBC-5) completes without error; and
2) an unexpected power removal occurs,

then, after power is restored, a read command that completes without error for an LBA included in that verify command shall return the logical block data that was accessed by that verify command before the unexpected power removal.

For sequential write required zones, if an unexpected power loss results in unwritten logical blocks in LBAs smaller than the write pointer for a zone and read operations access the unwritten logical blocks for that zone, then the device server shall terminate the operation with CHECK CONDITION status with sense key set to MEDIUM ERROR with additional sense code set to UNWRITTEN DATA IN ZONE.

After an unexpected power removal condition, the parameter data returned by a REPORT ZONES command (see 5.8) may contain write pointers that do not correspond to the ending LBA plus one of the last command to report completion prior to the unexpected power removal condition.

4.6.4 Media failure

If a host managed zoned block device has a media failure that affects one or more zones, then the device server may indicate that those zones are unavailable for reading or writing by changing the Zone Condition to OFFLINE.

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

a) WRITE ERROR - RECOVERY NEEDED;
b) WRITE ERROR - RECOVERY SCAN NEEDED; or
c) WRITE ERROR - INSUFFICIENT ZONE RESOURCES,

then the device server shall return the lowest numbered LBA where a media failure may have occurred during processing of that command in the INFORMATION field of the sense data (see SPC-6).

If the additional sense code is set to WRITE ERROR - RECOVERY SCAN NEEDED, then the application client should read all the logical blocks from the LBA indicated by the INFORMATION field to the LBA indicated by the write pointer minus one and take recovery actions for LBAs that return errors.

4.7 Interactions involving mode parameter block descriptors

If a zoned block device does not support changing its capacity by changing the NUMBER OF LOGICAL BLOCKS field in a mode parameter block descriptor (see SBC-5) using the MODE SELECT command (see SPC-6), then the device server responds to such requests as described in SBC-5.

4.8 Capacity reporting and LBAs out of range

The contents of the RETURNED LOGICAL BLOCK ADDRESS field in the READ CAPACITY (16) parameter data (see SBC-5) is indicated by the RC BASIS field (see table 16).

Table 16 – READ CAPACITY (16) parameter data as modified for zoned block devices

Bit / Byte	7	6	5	4	3	2	1	0
	⋮ (see SBC-5)							
12	(see SBC-5)		RC BASIS		(see SBC-5)			
	⋮ (see SBC-5)							

The RC BASIS field indicates the meaning of the value returned by the RETURNED LOGICAL BLOCK ADDRESS field in the READ CAPACITY (16) parameter data and is described in table 17.

Table 17 – RC BASIS field

Code	Description
00b	The RETURNED LOGICAL BLOCK ADDRESS field indicates: a) the highest LBA of a contiguous range of zones that are not sequential write required zones starting with the first zone, if the zoned block device is not a domains and realms zoned block device (see 4.2.2 and 4.2.3); and b) the highest LBA in zone domain 0 (see 4.2.4.3), if the zoned block device is a domains and realms zoned block device (see 4.2.4).
01b	The RETURNED LOGICAL BLOCK ADDRESS field indicates: a) the zoned maximum address, if the zoned block device is not a domains and realms zoned block device; and b) the highest LBA in zone domain 0 (see 4.2.4.3), if the zoned block device is a domains and realms zoned block device.
all others	Reserved

If the zoned block device is a domains and realms zoned block device, then:

a) the device server shall support accesses (e.g., reads and writes) to LBAs that are greater than the value in the RETURNED LOGICAL BLOCK ADDRESS field in the READ CAPACITY (16) parameter data and less than or equal to the zoned maximum address; and
b) the zoned maximum address shall not be affected by which zones are active zones or which zones are inactive zones.

Read commands, verify commands, and write commands that specify LBAs greater than the zoned maximum address shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

4.9 Constant zone starting LBA offsets

Some recording technologies do not have an efficient way to allow all zones to indicate the same zone length (i.e., set a constant value in the ZONE LENGTH field returned by the REPORT ZONES command (see 5.8)). However, many application clients depend upon having a constant offset between the zone starting LBAs.

A zoned block device (e.g., a host managed zone block device (see 4.2.3)) may set the ZONE ALIGNMENT METHOD field (see 6.5.2) to 8h to indicate that, for zones that are not gap zones (see 4.5.4) values returned by the REPORT ZONES command have:

a) a constant minimum difference between values in the ZONE STARTING LBA fields that is indicated by the ZONE STARTING LBA GRANULARITY field (see 6.5.2); and
b) the ZONE LENGTH field returned indicates the number of logical block that are capable of recording data in that zone.

The zoned block device accomplishes these objectives by paring each data recording zone with a gap zone that encompasses those logical blocks that are not capable of recording data (see figure 15).

	sequential write required zone (see 4.5.3.4)	zero or more gap zone	sequential write required zone	zero or more gap zone	
…	← zone length LBAs →	← fill LBAs →	← zone length LBAs →	← fill LBAs →	…
	← zone starting LBA granularity →		← zone starting LBA granularity →		

Figure 15 – Example of paired sequential write required zones and gap zones

For each zone that is not a gap zone, the number of LBAs in that zone (e.g., the zone length LBAs in a sequential write required zone as shown in figure 15) may be different from the number of LBAs in other zones that are not gap zones.

The sum of zone length LBAs and fill LBAs shall be equal to the value indicated by the ZONE STARTING LBA GRANULARITY field. If the number of zone length LBAs is equal to the zone starting LBA granularity, then the number of fill LBAs shall be zero and a gap zone shall not be present. The number of zone length LBAs shall not be greater than the zone starting LBA granularity.

4.10 Format operations

All of the functions defined for the FORMAT UNIT command (see SBC-5) and FORMAT WITH PRESET command (see SBC-5) are available for zoned block devices. A FORMAT UNIT command or FORMAT WITH PRESET command affects all zones, however some characteristics are dependent on the zone type (see 4.5) present in the zoned block device. A zoned block device may support multiple types of zones.

For each conventional zone (see 4.5.2), a format operation is performed as specified in SBC-5.

For each write pointer zone (see 4.5.3), a format operation is performed as specified in SBC-5 and a reset write pointer operation (see 4.5.3.4) is performed.

After completion of a format operation without error by the device server in a domains and realms zoned block device:

a) zones that were active zones before the format operation started shall be active zones; and
b) zones that were inactive zones before the format operation started shall be inactive zones.

A FORMAT UNIT command or FORMAT WITH PRESET command shall not perform a Reset Write Pointer operation on a write pointer zone that is associated with a Restrict Write Pointer Reset realm attribute (see 4.4.2) that is true.

As part of performing a format operation, the device server:

a) shall not change the contents of the ZONE STARTING LBA GRANULARITY field (see 6.5.2), unless that change is indicated by the DESIGNED ZONE STARTING LBA GRANULARITY field in the format preset descriptor (see SBC-5) specified by a FORMAT WITH PRESET command; and
b) may change the lengths of individual zones within the boundaries indicated by the ZONE STARTING LBA GRANULARITY field (see 4.9).

4.11 Sanitize operations

See SBC-5 for the definition of the SANITIZE command and sanitize operations. A sanitize operation affects all zones, however some characteristics are dependent on the presence of write pointer zones (see 4.5.3) in the zoned block device. As described in SBC-5, the ZNR bit in the SANITIZE command controls whether a reset write pointer operation (see 4.5.3.2.5) is performed on each write pointer zone as part of the sanitize operation.

For a CRYPTOGRAPHIC ERASE service action, a BLOCK ERASE service action, or an OVERWRITE action, the processing of the sanitize operation resulting from that service action:

a) shall affect the information on the logical unit's medium in a way that ensures that recovery of logical block data is not possible from:
   A) each conventional zone (see 4.5.2); and
   B) each write pointer zone (see 4.5.3);
b) shall include processing of the ZNR bit (see SBC-5) and associated reset write pointer operation, if any, if that sanitize operation:
   A) is successful; or
   B) fails in restricted completion mode (see SBC-5) and a subsequent EXIT FAILURE MODE service action is successful;

   and
c) successful completion of that sanitize operation shall include:
   A) completing the requirements described in SBC-5; and
   B) setting each write pointer zone's state (i.e., Zone Condition) to the initial state described in 4.5.3.6.1, unless:
      a) that write pointer zone was not able to be sanitized (e.g., due to an inability to write to that zone), in which case the zone's Zone Condition shall become OFFLINE; and
      b) that write pointer zone had a Zone Condition of READ ONLY before the sanitize operation was started and that zone was able to be sanitized, in which case the zone's Zone Condition:
         A) should remain READ ONLY; and
         B) may become OFFLINE for reasons that are outside the scope of this standard.

An OVERWRITE service action that completes without an error modifies the initialization pattern (see 4.5.3.1.4).

After completion of a sanitize operation without error by the device server in a domains and realms zoned block device:

a) zones that were active zones before the sanitize operation started shall be active zones; and
b) zones that were inactive zones before the sanitize operation started shall be inactive zones.

A SANITIZE command shall not perform a Reset Write Pointer operation on a write pointer zone that is associated with a Restrict Write Pointer Reset realm attribute (see 4.4.2) that is true.

4.12 Reservations

Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation. See SPC-6 for a description of reservations for commands described in that standard. See SBC-5 for a description of reservations for commands described in that standard. See table 18 for a description of reservations for commands described in this standard.

Commands from I_T nexuses holding a reservation should complete normally. Table 18 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 in table 18, this standard or SPC-6 defines the conditions that result in the device server completing the command with RESERVATION CONFLICT status.

Table 18 – ZBC-2 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
CLOSE ZONE	Conflict	Conflict	Allowed	Conflict	Conflict
FINISH ZONE	Conflict	Conflict	Allowed	Conflict	Conflict
OPEN ZONE	Conflict	Conflict	Allowed	Conflict	Conflict
REMOVE ELEMENT AND MODIFY ZONES	Conflict	Conflict	Allowed	Conflict	Conflict
REPORT REALMS	Allowed	Allowed	Allowed	Allowed	Allowed
REPORT ZONE DOMAINS	Allowed	Allowed	Allowed	Allowed	Allowed
REPORT ZONES	Allowed	Allowed	Allowed	Allowed	Allowed
RESET WRITE POINTER	Conflict	Conflict	Allowed	Conflict	Conflict
SEQUENTIALIZE ZONE	Conflict	Conflict	Allowed	Conflict	Conflict
ZONE ACTIVATE	Conflict	Conflict	Allowed	Conflict	Conflict
ZONE QUERY	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.

4.13 Caches

4.13.1 Caches overview

Zoned block devices may implement caches. A cache is an area of temporary storage in the zoned block device (e.g., to enhance performance) separate from the medium that is not directly accessible by the application client.

A cache stores logical block data.

A cache may be volatile or non-volatile. A volatile cache does not retain data through power cycles. A non-volatile cache retains data through power cycles. There may be a limit on the amount of time a non-volatile cache is able to retain data without power.

The cache model description for zone block devices is as specified in SBC-5 with additional requirements as specified in 4.13.

4.13.2 Write caching

While processing write commands, as a result of using write-back caching there is a period of time during which the logical block data may be lost if:

a) an unexpected power removal occurs (see 4.6.3); or
b) a hardware failure occurs.

If an error occurs during a write medium operation in a sequential write required zone and that error is reported as a deferred error, then the device server may invalidate cached logical block data for LBAs in that zone that are larger than the LBA reported with the deferred error. This invalidation may occur for data cached in both volatile and non-volatile caches.

4.13.3 Command interactions with caches

If the FUA bit is set to one in a read command or a write command that specifies LBAs in a sequential write required zone and any logical blocks in that zone are cached in a volatile cache, then the device server shall write all such logical blocks to the medium or non-volatile cache.

4.13.4 Write operation and write medium operation interactions with caches

When the device server performs a write cache operation that updates cached logical block data for a write pointer zone, the device server shall update the write pointer to reflect the completed write operation. If an unexpected power removal occurs (see 4.6.3), then upon the restoration of power, the value of every write pointer shall reflect the state of the medium and non-volatile cache, if any (e.g., a write pointer value may become smaller as a result of data lost from volatile cache).

4.13.5 Close zone and finish zone operation interactions with cache

If a zone's Zone Condition becomes CLOSED or FULL and any logical blocks in that zone are cached in a volatile cache, the device server writes those cached logical blocks to the medium or to non-volatile cache.

4.14 Interactions with WRITE LONG commands

For:

a) a sequential or before required zone (see 4.5.3.5);
b) a sequential write required zone (see 4.5.3.4); or
c) a sequential write preferred zone (see 4.5.3.3),

if:

1) a WRITE LONG command (see SBC-5) marks an LBA in that zone as uncorrectable; and
2) a reset write pointer operation is performed for that zone,

then for a subsequent read command or verify command for any such LBA that has not been written since the most recent time the Zone Condition was EMPTY, the device server shall process that command based on the value of the URSWRZ bit (see 6.5.2) as follows. If the URSWRZ bit is set to:

a) zero and the zone is not a sequential write preferred zone, then that command shall be terminated as described in SBC-5; and
b) one or the zone is a sequential write preferred zone, then
   A) the logical block data set to the most recent initialization pattern that was set as described in 4.5.3.1.4 shall be returned; and
   B) no data shall be written for a read command that has the FUA bit set to one (see SBC-5).

4.15 Interactions with storage element depopulation and restoration

4.15.1 Interactions with storage element depopulation and restoration operations that modify data

If the zoned block device is a domains and realms zoned block device (see 4.2.4), then upon completion of a storage element depopulation (see SBC-5) without error or a storage element restoration (see SBC-5) without error:

a) the data returned by the following commands may differ from data returned from the same command prior to command completion of that activity:
   A) the REPORT ZONES command (see 5.8);
   B) the REPORT ZONE DOMAINS command (see 5.7); and
   C) the REPORT REALMS command (see 5.6);
b) all of the zones in zone domain 0 are active; and
c) all of the zones in other zone domains are inactive.

If the zoned block device is a host aware zoned block device (see 4.2.2) or a host managed zoned block device (see 4.2.3), then upon completion of a storage element depopulation without error or a storage element restoration without error, the data returned by the REPORT ZONES command may differ from data returned from the same command prior to completion of that activity.

4.15.2 Storage element depopulation with zone modifications

4.15.2.1 Depopulation with zone modifications overview

Depopulation with zone modifications uses the REMOVE ELEMENT AND MODIFY ZONES command (see 5.5) to remove a storage element from service with LBAs that:

a) are not associated with that storage element not being affected; and
b) are associated with that storage element becoming:
   A) contained in a zone that has a Zone Condition of OFFLINE, if that zone is a write pointer zone (see 4.5.3); or
   B) associated with predicted unrecovered errors (see 4.15.2.3) and contained in a zone that has Predicted Unrecovered Errors Present (see 4.3.7) set to true, if that zone is a Conventional zone (see 4.5.2).

If the device server supports the REMOVE ELEMENT AND MODIFY ZONES command, then the device server shall support the RESTORE ELEMENTS AND REBUILD command (see SBC-5).

4.15.2.2 Depopulation with zone modifications processing

For each set of contiguous logical blocks that are not associated with the specified storage element by logical block access commands, the device server:

a) shall preserve all logical block data that is stored in those logical blocks;
b) shall not change any logical block provisioning information (see SBC-5) that is associated with those logical blocks;
c) shall not change the Zone Condition of the zone that contains those logical blocks; and
d) shall not change the value of Predicted Unrecovered Errors Present (see 4.3.7) for the zone that contains those logical blocks.

For each logical block that is associated with the specified storage element by logical block access commands, the device server shall:

a) cause the Zone Condition to become OFFLINE, if that logical block is contained in:
   A) a Sequential Write Preferred zone (see 4.5.3.3);
   B) a Sequential Write Required zone (see 4.5.3.4); or
   C) a Sequential Or Before Required zone (see 4.5.3.5);

b) if that logical block is contained in a Conventional zone (see 4.5.2), then establish future predicted unrecovered errors for that logical block or the set of contiguous logical blocks that contains that logical block as follows:
   A) cause Predicted Unrecovered Errors Present to become true for every zone that contains one or more of those logical blocks; and
   B) configure those logical blocks to invalidate cached data, if any, and to cause:
      a) subsequent read commands to result in predicted unrecovered read error processing (see 4.15.2.3.2); and
      b) subsequent write commands to result in predicted unrecovered write error processing (see 4.15.2.3.3);

   and
c) make no changes, if that logical block is contained in:
   A) a zone with a Zone Condition of INACTIVE, or
   B) a Gap zone (see 4.5.4).

Within a zone, the device server should process logical blocks from the smallest LBA in that zone to the largest LBA in that zone.

If during the processing described in this subclause (i.e., 4.15.2.2) the device server detects that the processing described in this subclause (i.e., 4.15.2.2) has already been performed, then device server shall not perform additional processing on the affected logical blocks and this shall not considered an error.

If the processing described in this subclause (i.e., 4.15.2.2) detects an error that prevents successful completion of that processing, then the processing of subsequent logical block access commands may be affected from the time that the device server detects that error until a subsequent REMOVE ELEMENT AND TRUNCATE command (see SBC-5) or a RESTORE ELEMENTS AND REBUILD command (see SBC-5) completes without error. If during this time interval a logical block access command specifies one or more LBAs that are affected by processing of the error, the device server may terminate that command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to INTERNAL TARGET FAILURE.

Upon the completion of the logical block processing described in this subclause (i.e., 4.15.2.2), the device server shall indicate that the specified storage element is depopulated (e.g., cause the parameter data returned by the GET PHYSICAL ELEMENT STATUS command (see SBC-5) PHYSICAL ELEMENT HEALTH field to contain FDh).

If the zoned block device is a domains and realms zoned block device (see 4.2.4), then the device server shall cause the Restrict Zone Activate realm attribute (see 4.4.3) to become true for each realm that contains a zone that is affected by the processing described in this subclause (i.e., 4.15.2.2).

From the time that processing of a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5) completes without error until the time that the processing described in this subclause (i.e., 4.15.2.2) completes, the device server shall provide pollable sense data (see SPC-6) with the sense key set to NO SENSE, the additional sense code set to:

a) DEPOPULATION INTERRUPTED, if the processing described in this subclause (i.e., 4.15.2.2) has been interrupted as described in 4.15.2.5; and
b) DEPOPULATION IN PROGRESS with the PROGRESS INDICATION field set to indicate the progress of the depopulation with zone modifications processing, if the processing described in this subclause (i.e., 4.15.2.2) has not been interrupted.

4.15.2.3 Handling unrecoverable errors

4.15.2.3.1 Handling unrecoverable errors overview

Depopulation with zone modifications processing shall not affect how the device server processes:

a) the rebuild assist mode (see SBC-5);
b) unpredicted unrecovered read errors (see SBC-5); or
c) unpredicted unrecovered write errors (see SBC-5).

If a zone has Predicted Unrecovered Errors Present (see 4.3.7) set to true, then the device server shall process logical block access commands that specify LBAs in that zone:

a) as described in 4.15.2.3.2 during reading logical block data from the medium; and
b) as described in 4.15.2.3.3 during writing logical block data to the medium.

4.15.2.3.2 Predicted unrecovered read errors in Conventional zones

If device server processes a read command that specifies one or more LBAs in a zone has Predicted Unrecovered Errors Present (see 4.3.7) set to true, and the device server detects a predicted unrecovered error, then the device server:

1) shall transfer to the Data-in Buffer the data for all logical blocks, if any, from the logical block specified as the starting LBA of that command up to the lowest numbered LBA for which a predicted unrecovered error has been established (see 4.15.2.2); and
2) shall terminate that command with CHECK CONDITION status with the sense key set to ABORTED COMMAND, the additional sense code set to MULTIPLE READ ERRORS, in addition to setting:
   A) the INFORMATION field (see SPC-6) to the logical block associated with the lowest numbered LBA for which data has not been transferred (i.e., the lowest numbered LBA for which a predicted unrecovered error has been established (see 4.15.2.2)); and
   B) the COMMAND-SPECIFIC INFORMATION field (see SPC-6) to the highest numbered LBA in the set of contiguous logical blocks, starting at the LBA specified in the INFORMATION field, for which predicted unrecovered errors have been established (see 4.15.2.2) without allowing the range of LBAs specified by that command to affect the contents of the COMMAND-SPECIFIC INFORMATION field.

If the application client receives sense data with the sense key set to ABORTED COMMAND, the additional sense code set to MULTIPLE READ ERRORS, and the INFORMATION field indicating a valid LBA (see SPC-6), then the application client should avoid sending logical block access commands that specify LBAs in the contiguous set of LBAs between the value in the INFORMATION field and the value in the COMMAND-SPECIFIC INFORMATION field.

4.15.2.3.3 Predicted unrecovered write errors in Conventional zones

If device server processes a write command that specifies one or more LBAs in a zone has Predicted Unrecovered Errors Present (see 4.3.7) set to true, and the device server detects a predicted unrecovered error, then the device server:

1) shall transfer from the Data-out Buffer the data for all logical blocks, if any, to the logical block specified as the starting LBA of that command up to the lowest numbered LBA for which a predicted unrecovered error has been established (see 4.15.2.2); and
2) shall terminate that command with CHECK CONDITION status with the sense key set to ABORTED COMMAND, the additional sense code set to MULTIPLE WRITE ERRORS, in addition to setting:
   A) the INFORMATION field (see SPC-6) to the logical block associated with the lowest numbered LBA for which data has not been transferred (i.e., the lowest numbered LBA for which a predicted unrecovered error has been established (see 4.15.2.2)); and
   B) the COMMAND-SPECIFIC INFORMATION field (see SPC-6) to the highest numbered LBA in the set of contiguous logical blocks, starting at the LBA specified in the INFORMATION field, for which predicted unrecovered errors have been established (see 4.15.2.2) without allowing the range of LBAs specified by that command to affect the contents of the COMMAND-SPECIFIC INFORMATION field.

If the application client receives sense data with the sense key set to ABORTED COMMAND, the additional sense code set to MULTIPLE WRITE ERRORS, and the INFORMATION field indicating a valid LBA (see SPC-6), then the application client should avoid sending logical block access commands that specify LBAs in the contiguous set of LBAs between the value in the INFORMATION field and the value in the COMMAND-SPECIFIC INFORMATION field.

4.15.2.4 Allowed commands during depopulation with zone modifications processing

After a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5) has returned command completion without error and before the device server has completed processing a depopulation with zone modifications (see 4.15.2.2), the device server shall process all commands based on whether depopulation with zone modifications have been made. If a command specifies access to a logical block and REMOVE ELEMENT AND MODIFY ZONES command processing has:

a) not modified access to that logical block, then the command shall be processed as if the REMOVE ELEMENT AND MODIFY ZONES command had not been received; and
b) has modified access to that logical block, then the command shall be processed based on the modifications that have been made.

For each logical block that is associated with the specified storage element for accesses to logical block data, the device server shall ensure that each subsequent command is processed in one of the two ways described in this subclause (i.e., 4.15.2.4) by maintaining synchronization between:

a) the processing described in 4.15.2.2, if any; and
b) the commands being processed.

If a command specifies one or more logical blocks that are in the process modified by the processing described in 4.15.2.2, then the device server may delay the processing of that command until those logical blocks have completed the processing described in 4.15.2.2.

4.15.2.5 Event handling actions

If the processing of depopulation with zone modifications (see 4.15.2.2) that is interrupted by a reset event (see SAM-6) or an I_T nexus loss even (see SAM-6), then the device server shall resume processing of depopulation with zone modifications upon completion actions that result from that event.

If the processing of depopulation with zone modifications that is interrupted by a power on event (see SAM-6), then the device server shall continue to accept commands as described in 4.15.2.4, except that:

a) the device server shall not perform any of the processing described in 4.15.2.2 until a REMOVE ELEMENT AND MODIFY ZONES command (see 5.5) has been processed without errors; and
b) while the device server is not performing the processing described in 4.15.2.2, a REQUEST SENSE command (see SPC-6) shall report that a depopulation with zone modifications has been interrupted (see 4.15.2.2).

5 Commands for zoned block devices

5.1 Commands for zoned block devices overview

5.1.1 Summary of commands for zoned block devices

For zoned block devices, the contents of the PERIPHERAL DEVICE TYPE field in the standard INQUIRY data (see SPC-6) indicate the command support requirements as described in 4.1.2.

Table 19 summarizes the commands that are unique to zoned block devices.

Table 19 – Summary of commands that are unique to zoned block devices

Command	Operation code [a]	Table 20 CDB format [b]	Reference
CLOSE ZONE	94h/01h	Yes	5.2
FINISH ZONE	94h/02h	Yes	5.3
OPEN ZONE	94h/03h	Yes	5.4
REMOVE ELEMENT AND MODIFY ZONES	9Eh/1Ah	No	5.5
REPORT REALMS	95h/06h	No	5.6
REPORT ZONE DOMAINS	95h/07h	No	5.7
REPORT ZONES	95h/00h	No	5.8
RESET WRITE POINTER	94h/04h	Yes	5.9
SEQUENTIALIZE ZONE	94h/10h	Yes	5.10
ZONE ACTIVATE	95h/08h	No	5.11
ZONE QUERY	95h/09h	No	5.12

[a] For operation codes 94h and 95h, combinations of operation code and service action not shown in this table are reserved.

[b] Yes means that the CDB format described in 5.1.2 is used. No means that a CDB format other than the one described in 5.1.2 is used.

5.1.2 Zoned block device 16-byte CDB format with no data transfer

Table 20 shows the typical format of a 16-byte CDB for some zoned block device commands (see table 19). The CDB format shown in table 20 includes common locations for:

a) the ZONE ID field;
b) the ZONE COUNT field; and
c) the ALL bit.

Table 20 – Typical 16-byte zoned block device CDB format with no data transfer

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (94h)							
1	Command specific fields, if any			SERVICE ACTION				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10								
11	Command specific fields, if any							
12	(MSB)							
13	ZONE COUNT							(LSB)
14	Command specific fields, if any							ALL
15	CONTROL							

The OPERATION CODE field, SERVICE ACTION field, CONTROL byte, and command specific fields are defined by each zoned block device command.

The usage of the ZONE ID field and the ZONE COUNT field depends on the value of the ALL bit.

If the ALL bit is set to zero, then:

a) the ZONE ID field specifies the lowest LBA of a write pointer zone;
b) if the ZONE COUNT field is set to:
   A) zero or one, then the command requests the device server to perform the requested operation on the zone specified by the ZONE ID field; and
   B) a value greater than one, then the command requests the device server to perform the requested operation on the zone range specified by the ZONE ID field and the ZONE COUNT field;

   and
c) the device server shall terminate the command with CHECK CONDITION status:
   A) with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE, if:
      a) the ZONE ID field specifies an LBA that is greater than the zoned maximum address; or
      b) the specified zone range, if any, includes an LBA that is greater than the zoned maximum address;
   B) with the sense key set to DATA PROTECT and the additional sense code set to ZONE IS READ ONLY, if the ZONE COUNT field is set to zero and the ZONE ID field specifies zone with a Zone Condition of READ ONLY;
   C) with the sense key set to DATA PROTECT and the additional sense code set to ZONE IS OFFLINE, if the ZONE COUNT field is set to zero and the ZONE ID field specifies zone with a Zone Condition OFFLINE;
   D) with the sense key set to DATA PROTECT and the additional sense code set to ZONE IS INACTIVE, if the ZONE COUNT field is set to zero and the ZONE ID field specifies zone with a Zone Condition INACTIVE;

E) with the sense key set to ILLEGAL REQUEST and the additional sense code set to ATTEMPT TO ACCESS GAP ZONE, if:
   a) the ZONE ID field specifies an LBA that is in a gap zone (see 4.5.4); or
   b) the specified zone range, if any, includes a gap zone;
F) with the sense key and additional sense code set to the values specified for any command specific errors, see:
   a) 5.2 for the CLOSE ZONE command;
   b) 5.3 for the FINISH ZONE command;
   c) 5.4 for the OPEN ZONE command;
   d) 5.9 for the RESET WRITE POINER command; and
   e) 5.10 for the SEQUENTIALIZE ZONE command;
   and
G) with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB, if:
   a) the ZONE ID field:
      A) does not specify the lowest LBA of a zone; or
      B) specifies a conventional zone;
   or
   b) the specified zone range, if any, includes:
      A) a zone that is a conventional zone; or
      B) more than one zone domain (see 4.2.4.2).

If the ALL bit set to one, then:

a) the command requests the device server to:
   A) ignore the contents of the ZONE ID field; and
   B) perform the requested operation as defined for the command, if the ZONE COUNT field is set to zero; and
b) if the ZONE COUNT field is not set to zero, then the device server shall terminated 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.

5.2 CLOSE ZONE command

The CLOSE ZONE command (see table 21) requests the device server to perform close zone operations (see 4.5.3.2.3).

Table 21 – CLOSE ZONE command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (94h)							
1	Reserved			SERVICE ACTION (01h)				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10	Reserved							
11								
12	(MSB)							
13	ZONE COUNT							(LSB)
14	Reserved							ALL
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 21 for the CLOSE ZONE command.

The ZONE ID field, ZONE COUNT field, and ALL bit are defined in 5.1.2.

For a domains and realms zoned block device (see 4.2.4), 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, if the ALL bit is set to zero, and:

a) the ZONE ID field specifies an LBA that is in a sequential or before required zone (see 4.5.3.5); or
b) the specified zone range (see 5.1.2), if any, includes a sequential or before required zone.

The CONTROL byte is defined in SAM-6.

If the CLOSE ZONE command is not terminated with CHECK CONDITION status (see 5.1.2), then the device server processes the command as described in table 22.

Table 22 – CLOSE ZONE command processing

ZONE COUNT field	ALL bit	Description
zero or one	zero	If the specified zone has a Zone Condition of: a) EMPTY, CLOSED, or FULL, then the device server shall make no changes in the Zone Condition and not return an error; and b) IMPLICITLY OPENED or EXPLICITLY OPENED, then the device server shall perform a close zone operation (see 4.5.3.2.3) on the specified zone.
two or more	zero	For each zone in the specified zone range that has a Zone Condition of IMPLICITLY OPENED or EXPLICITLY OPENED, the device server shall perform a close zone operation.
zero	one	For each zone that has a Zone Condition of IMPLICITLY OPENED or EXPLICITLY OPENED, the device server shall perform a close zone operation.
non-zero	one	See 5.1.2.

5.3 FINISH ZONE command

The FINISH ZONE command (see table 23) requests the device server to perform finish zone operations (see 4.5.3.2.4).

The device server returns the initialization pattern for all unwritten LBAs in this zone (see 4.5.3.3, 4.5.3.4, and 4.5.3.5) in response to a read operation. The device server may write the initialization pattern to the media for unwritten LBAs.

Table 23 – FINISH ZONE command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (94h)							
1	Reserved			SERVICE ACTION (02h)				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10				Reserved				
11								
12	(MSB)							
13	ZONE COUNT							(LSB)
14	Reserved							ALL
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 23 for the FINISH ZONE command.

The ZONE ID field, ZONE COUNT field, and ALL bit are defined in 5.1.2.

For a domains and realms zoned block device (see 4.2.4), 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, if the ALL bit is set to:

a) zero and:
   A) the ZONE ID field specifies an LBA that is in a sequential or before required zone (see 4.5.3.5); or
   B) the specified zone range (see 5.1.2), if any, includes a sequential or before required zone;
   and
b) one and at least one sequential or before required zone has a Zone Condition of IMPLICITLY OPENED.

The CONTROL byte is defined in SAM-6.

If the FINISH ZONE command is not terminated with CHECK CONDITION status (see 5.1.2), then the device server processes the command as described in table 24.

Table 24 – FINISH ZONE command processing

ZONE COUNT field	ALL bit	Description
zero or one	zero	If the specified zone has a Zone Condition of: a) FULL, then the device server shall make no changes in the Zone Condition and not return an error; and b) IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or EMPTY, then the device server shall perform a finish zone operation (see 4.5.3.2.4) on the specified zone.
two or more	zero	For each zone in the specified zone range that has a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, or CLOSED, the device server shall perform a finish zone operation.
zero	one	For each zone that has a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, or CLOSED, the device server shall perform a finish zone operation.
non-zero	one	See 5.1.2.

5.4 OPEN ZONE command

The OPEN ZONE command (see table 25) requests the device server to perform open zone operations (see 4.5.3.2.2).

Table 25 – OPEN ZONE command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (94h)							
1	Reserved			SERVICE ACTION (03h)				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10	Reserved							
11								
12	(MSB)							
13	ZONE COUNT							(LSB)
14	Reserved							ALL
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 25 for the OPEN ZONE command.

The ZONE ID field, ZONE COUNT field, and ALL bit are defined in 5.1.2.

The device server shall terminate the command with CHECK CONDITION status, with the sense key set to DATA PROTECT and the additional sense code set to INSUFFICIENT ZONE RESOURCES, if the value in the MAXIMUM NUMBER OF OPEN SEQUENTIAL WRITE REQUIRED ZONES field (see 6.5.2) is less than or equal to the sum of:

a) the number of sequential write required zones (see 4.5.3.4) in the logical unit with a Zone Condition of EXPLICITLY OPENED; and
b) the number of sequential write required zones in:
   A) the specified zone range (see 5.1.2) with a Zone Condition of IMPLICITLY OPENED, CLOSED, or EMPTY, if the ALL bit is set to zero; or
   B) the logical unit with a Zone Condition of CLOSED, if the ALL bit is set to one.

For a domains and realms zoned block device (see 4.2.4), 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, if the ALL bit is set to zero, and:

a) the ZONE ID field specifies an LBA that is in a sequential or before required zone (see 4.5.3.5); or
b) the specified zone range (see 5.1.2), if any, includes a sequential or before required zone.

The CONTROL byte is defined in SAM-6.

If the OPEN ZONE command is not terminated with CHECK CONDITION status (see 5.1.2), then the device server processes the command as described in table 26.

Table 26 – OPEN ZONE command processing

ZONE COUNT field	ALL bit	Description
zero or one	zero	If the specified zone has a Zone Condition of: a) EXPLICITLY OPENED or FULL, then the device server shall make no changes in the Zone Condition and not return an error; and b) IMPLICITLY OPENED, CLOSED, or EMPTY, then the device server shall perform an open zone operation (see 4.5.3.2.2) on the specified zone.
two or more	zero	For each zone in the specified zone range that has a Zone Condition of IMPLICITLY OPENED, CLOSED, or EMPTY, the device server shall perform an open zone operation.
zero	one	For each zone that has a Zone Condition of CLOSED, the device server shall perform an open zone operation.
non-zero	one	See 5.1.2.

5.5 REMOVE ELEMENT AND MODIFY ZONES command

The REMOVE ELEMENT AND MODIFY ZONES command (see table 27) requests that the device server perform storage element depopulation with zone modifications (see 4.15.2 and SBC-5). For additional information about storage element depopulation, see SBC-5.

If deferred microcode has been saved and not activated (see SPC-6), then the device server shall terminate this command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, MICROCODE ACTIVATION REQUIRED.

Table 27 – REMOVE ELEMENT AND MODIFY ZONES command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Eh)							
1	Reserved			SERVICE ACTION (1Ah)				
2	Reserved							
•••								
9								
10	(MSB)							
•••	ELEMENT IDENTIFIER							
13								(LSB)
14	Reserved							
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 27 for the REMOVE ELEMENT AND MODIFY ZONES command.

The ELEMENT IDENTIFIER field specifies the element identifier associated with the storage element (see SBC-5) to be depopulated. 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, if the ELEMENT IDENTIFIER field specifies a physical element:

a) that is not supported by the device server (see SBC-5);
b) for which the device server does not support depopulation; or
c) that is not a storage element, (i.e., the PHYSICAL ELEMENT TYPE field in the GET PHYSICAL ELEMENT STATUS parameter data (see SBC-5) is not set to 01h in the corresponding physical element status descriptor).

If the ELEMENT IDENTIFIER field specifies an element identifier that is associated with a storage element that is depopulated, then the device server shall make no changes and not consider this an error.

The CONTROL byte is defined in SAM-6.

After successful command completion, further storage element depopulation with zone modifications processing may occur as described in 4.15.2 and SBC-5.

5.6 REPORT REALMS command

5.6.1 REPORT REALMS command overview

The REPORT REALMS command (see table 28) requests that the device server transfer parameter data describing the realms structure (see 5.6.2) of the zoned block device.

Table 28 – REPORT REALMS command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (95h)							
1	Reserved			SERVICE ACTION (06h)				
2	(MSB)							
•••	REALM LOCATOR							
9								(LSB)
10	(MSB)							
•••	ALLOCATION LENGTH							
13								(LSB)
14	Reserved		REPORTING OPTIONS					
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 28 for the REPORT REALMS command.

The REALM LOCATOR field specifies information that locates the first realm to be reported. The first realm descriptor (see 5.6.2.2) shall describe:

a) the realm that contains the LBA specified by the REALM LOCATOR field, if that realm meets the criteria specified by the REPORTING OPTIONS field; or
b) the first realm, if any, that:
   A) meets the criteria specified by the REPORTING OPTIONS field; and
   B) contains an LBA that is greater than the value in the REALM LOCATOR field.

If the REALM LOCATOR field specifies an LBA that is greater than the zoned maximum address, 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 LOGICAL BLOCK ADDRESS OUT OF RANGE.

The ALLOCATION LENGTH field is defined in SPC-6. If the allocation length is less than 64, 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 REPORTING OPTIONS field (see table 29) specifies the information to be returned in the parameter data.

Table 29 – REPORT REALMS REPORTING OPTIONS field

Code	Description
00h	Report all realms
01h	Report all realms that contain Sequential Or Before Required zones that are active zones
02h	Report all realms that contain Sequential Write Required zones that are active zones
03h	Report all realms that contain Sequential Write Preferred zones that are active zones
all others	Reserved

The CONTROL byte is defined in SAM-6.

5.6.2 REPORT REALMS parameter data

5.6.2.1 REPORT REALMS parameter data overview

The REPORT REALMS parameter data (see table 30) contains information about the specified zone domains starting with the zone domain specified by the ZONE DOMAIN LOCATOR field.

Table 30 – REPORT REALMS parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
•••				REALMS DESCRIPTORS LIST LENGTH (n–63)				
3								(LSB)
4	(MSB)							
•••				REALMS COUNT				
7								(LSB)
8	(MSB)							
•••				REALMS DESCRIPTOR LENGTH				
11								(LSB)
12	(MSB)							
•••				NEXT REALM LOCATOR				
19								(LSB)
20								
•••				Reserved				
63								
Realm descriptors list								
64								
•••				Realm descriptor [first] (see 5.6.2.2)				
63+r [a]								
				⋮				
n–1–r [a]								
•••				Realm descriptor [last] (see 5.6.2.2)				
n								

[a] Where r equals the contents of the REALMS DESCRIPTOR LENGTH field.

The REALMS DESCRIPTORS LIST LENGTH field indicates the number of bytes in the realm descriptors list.

The REALMS COUNT field indicates the number of realms configured on the device server. The value in the REALMS COUNT field shall not be affected by:

a) the contents of the REPORTING OPTIONS field (see 5.6.1);
b) the contents of the REALM LOCATOR field (see 5.6.1); or
c) the contents of or number of realms descriptors returned in the REPORT REALMS parameter data.

The REALMS DESCRIPTOR LENGTH field indicates the number of bytes in one realm descriptor (see 5.6.2.2).

The NEXT REALMS LOCATOR field indicates a possible contents for a subsequent REPORT REALMS command as follows. If the parameter data transferred to the application client by this REPORT REALMS command:

a) contains the last realm descriptor requested by the REPORT REALMS command (see 5.6.1), then the NEXT REALM LOCATOR field shall be set to zero; and
b) does not contain the last realm descriptor requested by the REPORT REALMS command, then a subsequent REPORT REALMS command should return data that begins reporting realms where this REPORT REALMS command stopped, if that subsequent REPORT REALMS command has:
   A) the REALM LOCATOR field (see 5.6.1) set to the value in this NEXT REALM LOCATOR field; and
   B) the REPORTING OPTIONS field (see 5.6.1) set to the same value as the REPORTING OPTIONS field of this REPORT REALMS command.

The realm descriptors contain information about the specified realms as defined in 5.6.2.2.

5.6.2.2 Realm descriptor

5.6.2.2.1 Realm descriptor overview

Each realm descriptor (see table 31) contains information about one realm and all the zone domains included in that realm. The contents of the REALM LOCATOR field (see 5.6.1) shall not affect the zone domains included in any realm descriptor. The realm descriptors shall be sorted in ascending order based on the contents of the REALM ID field.

Table 31 – Realm descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
•••				REALM ID				
3								(LSB)
4	(MSB)							
5				REALM RESTRICTIONS				(LSB)
6	Reserved							
7	ACTIVE ZONE DOMAIN ID							
8								
•••				Reserved				
15								
Realm Start/End descriptors list								
16								
•••				Realm Start/End descriptor for zone domain 0				
31								
32								
•••				Realm Start/End descriptor for zone domain 1				
47								
48								
•••				Realm Start/End descriptor for zone domain 2, if any				
63								
				⋮				
n–15								
•••				Realm Start/End descriptor for zone domain zd [a], if any				
n								

[a] Where zd is one less than the value in the NUMBER OF SUPPORTED ZONE DOMAINS field in the parameter data returned by the REPORT ZONE DOMAINS command (see 5.7).

The REALM ID field indicates the identifier number assigned to this realm. Realms are numbered from 0 to n incrementing by one for each realm.

The REALM RESTRICTIONS field (see table 32) indicates what restrictions, if any, affect this realm.

Table 32 – REALM RESTRICTIONS field

Code	Description
0000h	No restrictions affect this realm.
0001h	The Restrict Zone Activate realm attribute (see 4.4.3) is true.
0003h	The Restrict Zone Activate realm attribute (see 4.4.3) is true and the Restrict Write Pointer Reset realm attribute (see 4.4.2) is true.
all others	Reserved

The ACTIVE ZONE DOMAIN ID field indicates the zone domain ID (see 4.2.4.2) of the zone domain that contains active zones in this realm. If no zone domain contains active zones in this realm, the ACTIVE ZONE DOMAIN ID field shall contain FFh.

5.6.2.2.2 Realm Start/End descriptor

Each realm start/end descriptor (see table 33) contains information about one zone domain that is in the realm described by this realm descriptor (see 5.6.2.2.2).

Table 33 – Realm Start/End descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	REALM STARTING LBA							
•••								
7								
8	REALM ENDING LBA							
•••								
15								

The REALM STARTING LBA field shall contain the lowest LBA in the realm described by this realm start/end descriptor The processing of a ZONE ACTIVATE command (see 5.11) may result in the contents of the REALM STARTING LBA field changing (see 4.2.4.7).

The REALM ENDING LBA field shall contain the highest LBA in the realm described by this realm start/end descriptor. A REALM ENDING LBA field set to zero indicates that the zone domain associated with this realm start/end descriptor is not contained in the realm described by this realm descriptor The processing of a ZONE ACTIVATE command may result in the contents of the REALM ENDING LBA field changing (see 4.2.4.7).

5.7 REPORT ZONE DOMAINS command

5.7.1 REPORT ZONE DOMAINS command overview

The REPORT ZONE DOMAINS command (see table 34) requests that the device server transfer parameter data describing the zone domains structure (see 5.7.2) of the zoned block device.

Table 34 – REPORT ZONE DOMAINS command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (95h)							
1	Reserved			SERVICE ACTION (07h)				
2	(MSB)							
•••	ZONE DOMAIN LOCATOR							
9								(LSB)
10	(MSB)							
•••	ALLOCATION LENGTH							
13								(LSB)
14	Reserved			REPORTING OPTIONS				
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 34 for the REPORT ZONE DOMAINS command.

The ZONE DOMAIN LOCATOR field specifies information that locates the first zone domain to be reported. The first zone domain descriptor (see 5.7.2) shall describe:

a) the zone domain that contains the LBA specified the ZONE DOMAIN LOCATOR field, if that zone domain meets the criteria specified by the REPORTING OPTIONS field; or
b) the first zone domain, if any, that:
   A) meets the criteria specified by the REPORTING OPTIONS field; and
   B) contains an LBA that is greater than the value in the ZONE DOMAIN LOCATOR field.

If the ZONE DOMAIN LOCATOR field specifies an LBA that is greater than the zoned maximum address, 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 LOGICAL BLOCK ADDRESS OUT OF RANGE.

The ALLOCATION LENGTH field is defined in SPC-6. If the allocation length is less than 160, 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 REPORTING OPTIONS field (see table 35) specifies the information to be returned in the parameter data.

Table 35 – REPORT ZONE DOMAINS REPORTING OPTIONS field

Code	Description
00h	Report all zone domains
01h	Report all zone domains in which all zones are active
02h	Report all zone domains that contain active zones
03h	Report all zone domains that do not contain any active zones
all others	Reserved

The CONTROL byte is defined in SAM-6.

5.7.2 REPORT ZONE DOMAINS parameter data

The REPORT ZONE DOMAINS parameter data (see table 36) contains information about the specified zone domains starting with the zone domain specified by the ZONE DOMAIN LOCATOR field.

Table 36 – REPORT ZONE DOMAINS parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
•••				ZONE DOMAINS LIST LENGTH (n–63)				
3								(LSB)
4	(MSB)							
•••				ZONE DOMAINS RETURNED LIST LENGTH				
7								(LSB)
8	NUMBER OF ZONE DOMAINS SUPPORTED							
9	ZONE DOMAINS REPORTED							
10	REPORTING OPTIONS							
11								
•••				Reserved				
15								
16	(MSB)							
•••				ZONE DOMAIN LOCATOR				
23								(LSB)
24								
•••				Reserved				
63								
Zone domain descriptors list								
64								
•••				Zone domain descriptor [first]				
159								
				⋮				
n–95								
•••				Zone domain descriptor [last]				
n								

The ZONE DOMAINS LIST LENGTH field indicates the length in bytes of the zone domain descriptors list. The contents of the ZONE DOMAINS LIST LENGTH field shall not altered based on the allocation length (see SPC-6).

The ZONE DOMAINS RETURNED LIST LENGTH field indicates the length in bytes of the part of the zone domain descriptors list that are being returned for the application client to process. If the allocation length (see SPC-6) is not sufficient to contain all of the contents of the zone domain descriptors list, then the ZONE DOMAINS RETURNED LIST LENGTH field shall be altered to indicate the truncation of the zone domain descriptors list.

The NUMBER OF ZONE DOMAINS SUPPORTED field indicates the total number of zone domains supported by the device server, regardless of the value of the REPORTING OPTIONS field.

The ZONE DOMAINS REPORTED field indicates the number of zone domain descriptors returned based on the REPORTING OPTIONS field and the ZONE DOMAIN LOCATOR field.

The REPORTING OPTIONS field in the parameter data is a copy of the REPORTING OPTIONS field in the REPORT ZONE DOMAINS command (see 5.7.1).

The ZONE DOMAIN LOCATOR field in the parameter data is a copy of the ZONE DOMAIN LOCATOR field in the REPORT ZONE DOMAINS command (see 5.7.1).

Each zone domain descriptor (see table 37) describes one zone domain (see 4.2.4.2). Zone domain descriptors shall be sorted in order of increasing value of the ZONE DOMAIN ID field.

Table 37 – Zone domain descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	ZONE DOMAIN ID							
1	Reserved							
•••								
15								
16	(MSB)							
•••	ZONE COUNT							
23								(LSB)
24	(MSB)							
•••	STARTING LBA							
31								(LSB)
32	(MSB)							
•••	ENDING LBA							
39								(LSB)
40	ZONE DOMAIN ZONE TYPE							
41	Reserved							
42	Reserved						VZDZT	SRB
43	Reserved							
•••								
95								

The ZONE DOMAIN ID field indicates the zone domain ID (see 4.2.4.2) associated with this zone domain.

The ZONE COUNT field indicates the number of zones in this zone domain.

The STARTING LBA field indicates the starting LBA for this zone domain.

The ENDING LBA field indicates the ending LBA for this zone domain.

The contents of the ZONE DOMAIN ZONE TYPE field depend on the value of the VZDZT bit. If the VZDZT bit is set to:

a) zero, then:
   A) the Zone Type (see 4.3.2) for all zones in this zone domain may or may not be the same; and
   B) the contents of the ZONE DOMAIN ZONE TYPE field are invalid;
   and
b) one, then:
   A) the Zone Type attribute for all zones in this zone domain is the same; and
   B) the ZONE DOMAIN ZONE TYPE field indicates that zone type as described in table 43.

If the valid zone domain zone type (VZDZT) bit is set to:

a) zero, then the contents of the ZONE DOMAIN ZONE TYPE field are invalid; and
b) one, then the contents of the ZONE DOMAIN ZONE TYPE field are valid.

An shifting realm boundaries (SRB) bit set to zero indicates that realm boundaries do not shift during the processing of a ZONE ACTIVATE command (see 5.11). An SRB bit set to one indicates that realm boundaries may shift during the processing of a ZONE ACTIVATE command. Other realm boundary considerations are described in 4.2.4.7.

5.8 REPORT ZONES command

5.8.1 REPORT ZONES command overview

The REPORT ZONES command (see table 38) requests that the device server transfer parameter data describing the zone structure of the zoned block device.

Table 38 – REPORT ZONES command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (95h)							
1	Reserved			SERVICE ACTION (00h)				
2	(MSB)							
•••	ZONE START LBA							
9								(LSB)
10	(MSB)							
•••	ALLOCATION LENGTH							
13								(LSB)
14	PARTIAL	Reserved		REPORTING OPTIONS				
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 38 for the REPORT ZONES command.

The ZONE START LBA field specifies an LBA in the first zone to be reported. If the ZONE START LBA field does not specify the lowest LBA of a zone, then the device server uses the lowest LBA of the zone that contains the specified LBA to specify the first zone to be reported. If the ZONE START LBA field specifies an LBA that is greater than the value in the MAXIMUM LBA field (see table 42), 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 LOGICAL BLOCK ADDRESS OUT OF RANGE.

The ALLOCATION LENGTH field is defined in SPC-6.

The PARTIAL bit modifies the definition of the ZONE LIST LENGTH field as described in 5.8.2.

The REPORTING OPTIONS field (see table 39) specifies the information to be returned in the parameter data.

Table 39 – REPORT ZONES REPORTING OPTIONS field

Code	Description
00h	List all of the zones in the zoned block device.
01h	List the zones with a Zone Condition of EMPTY.
02h	List the zones with a Zone Condition of IMPLICITLY OPENED.
03h	List the zones with a Zone Condition of EXPLICITLY OPENED.
04h	List the zones with a Zone Condition of CLOSED.
05h	List the zones with a Zone Condition of FULL.
06h	List the zones with a Zone Condition of READ ONLY.
07h	List the zones with a Zone Condition of OFFLINE.
08h	List the zones with a Zone Condition of INACTIVE.
09h to 0Fh	Reserved
10h	List of the zones with RWP Recommended (see 4.3.5) set to true.
11h	List of the zones with Non-Sequential Write Resources Active (see 4.3.6) set to true.
12h to 3Dh	Reserved
3Eh	List all of the zones in the zoned block device, except zones with a Zone Type of GAP.
3Fh	List of the zones with a Zone Condition of NOT WRITE POINTER.

The CONTROL byte is defined in SAM-6.

5.8.2 REPORT ZONES parameter data

The REPORT ZONES parameter data is defined in table 40.

Table 40 – REPORT ZONES parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
•••	ZONE LIST LENGTH (n–63)							
3								(LSB)
4	Reserved				SAME			
5								
•••	Reserved							
7								
8	(MSB)							
•••	MAXIMUM LBA							
15								(LSB)
16	(MSB)							
•••	REPORTED ZONE STARTING LBA GRANULARITY							
23								(LSB)
24								
•••	Reserved							
63								
Zone descriptors list								
64								
•••	Zone descriptor [first]							
127								
	⋮							
n–63								
•••	Zone descriptor [last]							
n								

The ZONE LIST LENGTH field shall contain the length in bytes of the zone descriptors list. The zone descriptors list is the list of zones that:

a) meet the requirements of the REPORTING OPTIONS field; and
b) include the LBA specified by the ZONE START LBA field or have a lowest LBA that is greater than the LBA specified by the ZONE START LBA field.

If the PARTIAL bit (see 5.8.1) is set to zero, then the content of the ZONE LIST LENGTH field is not altered based on the allocation length (see SPC-6). If the PARTIAL bit is set to one then the ZONE LIST LENGTH field shall be set to the lesser of:

a) the allocation length minus 64 if the allocation length is greater than 64;
b) zero if the allocation length is less than or equal to 64; or
c) the length of the zone descriptors list.

The SAME field is defined in table 41. If the ZONE LIST LENGTH field is zero then the SAME field is invalid and should be ignored by the application client. If the last zone descriptor is an incomplete zone descriptor (e.g., as a result of the PARTIAL bit being set to one and the allocation length not being a multiple of 64), then that zone descriptor shall be included in the determination of the value of the SAME field.

Table 41 – SAME field description

Code	Description
0h	The zone type and zone length in each zone descriptor may be different.
1h	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 in the zone descriptor list.
2h	The zone type in each zone descriptor is equal to the zone type indicated in the first zone descriptor in the zone descriptor list. The zone length of each zone except the last zone is equal to the zone length of the first zone descriptor in the zone descriptor list. The zone length of the last zone descriptor is different than the zone length of the first descriptor in the zone descriptor list.
3h	The zone type in each descriptor may be different. The zone length in each zone descriptor is equal to the zone length indicated in the first zone descriptor in the zone descriptor list.
4h to Fh	Reserved

The MAXIMUM LBA field contains the LBA of the last logical block on the logical unit.

If the ZONE ALIGNMENT METHOD field (see 6.5.2) is:

a) set to zero, then the REPORTED ZONE STARTING LBA GRANULARITY field shall be set to zero; and
b) not set to zero, then the REPORTED ZONE STARTING LBA GRANULARITY field shall be set to the value contained in the ZONE STARTING LBA GRANULARITY field (see 6.5.2).

The zone descriptors list contains zone descriptors that shall be sorted in ascending order based on the ZONE START LBA field of each zone descriptor.

Each zone descriptor (see table 42) contains the description of a single zone.

Table 42 – Zone descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved	PUEP	NON_SEQ	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 Zone Type (see 4.3.2) of the zone as described in table 43.

Table 43 – Zone descriptor ZONE TYPE field

Code	Zone Type	Reference
0h	Reserved	
1h	CONVENTIONAL	4.5.2
2h	SEQUENTIAL WRITE REQUIRED	4.5.3.4
3h	SEQUENTIAL WRITE PREFERRED	4.5.3.3
4h	SEQUENTIAL OR BEFORE REQUIRED	4.5.3.5
5h	GAP	4.5.4
6h to Fh	Reserved	

The ZONE CONDITION field indicates the Zone Condition (see 4.3.3) of the zone as described in table 44.

Table 44 – Zone descriptor ZONE CONDITION field

Code	Zone Condition	The content of the WRITE POINTER LBA field is valid	Reference
0h	NOT WRITE POINTER	No	
1h	EMPTY	Yes	4.5.3.6.2
2h	IMPLICITLY OPENED	Yes	4.5.3.6.3
3h	EXPLICITLY OPENED	Yes	4.5.3.6.4
4h	CLOSED	Yes	4.5.3.6.5
5h	INACTIVE	No	4.5.3.6.9
6h to Ch	Reserved		
Dh	READ ONLY	No	4.5.3.6.7
Eh	FULL	No	4.5.3.6.6
Fh	OFFLINE	No	4.5.3.6.8

The value of the predicted unrecovered errors present (PUEP) bit is based on Predicted Unrecovered Errors Present (see 4.3.7). If Predicted Unrecovered Errors Present is:

a) false, then the PUEP bit shall be set to zero; or
b) true, then the PUEP bit shall be set to one.

The value of the non-sequential (NON_SEQ) bit is based on Non-Sequential Write Resources Active (see 4.3.6). If Non-Sequential Write Resources Active is:

a) false, then the NON_SEQ bit shall be set to zero; or
b) true, then the NON_SEQ bit shall be set to one.

The value of the RESET bit is based on RWP Recommended (see 4.3.5). If RWP Recommended is:

a) false, then the RESET bit shall be set to zero; or
b) true, then the RESET bit shall be set to one.

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

The ZONE START LBA field indicates the lowest 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 (i.e., the write pointer). The content of the WRITE POINTER LBA field is invalid if the content of the ZONE CONDITION field (see table 44) indicates that the WRITE POINTER LBA field is invalid.

5.9 RESET WRITE POINTER command

The RESET WRITE POINTER command (see table 45) requests the device server to perform reset write pointer operations (see 4.5.3.2.5).

Table 45 – RESET WRITE POINTER command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (94h)							
1	Reserved			SERVICE ACTION (04h)				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10	Reserved							
11								
12	(MSB)							
13	ZONE COUNT							(LSB)
14	Reserved							ALL
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 45 for the RESET WRITE POINTER command.

The ZONE ID field, ZONE COUNT field, and ALL bit are defined in 5.1.2.

For a domains and realms zoned block device (see 4.2.4), 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 RESET WRITE POINTER NOT ALLOWED, if:

a) the ZONE ID field specifies an LBA that is in a zone that is associated with a Restrict Write Pointer Reset realm attribute (see 4.4.2) that is true; or
b) the specified zone range, if any, includes a zone that is associated with a Restrict Write Pointer Reset realm attribute that is true.

The CONTROL byte is defined in SAM-6.

If the RESET WRITE POINTER command is not terminated with CHECK CONDITION status (see 5.1.2), then the device server processes the command as described in table 46.

Table 46 – RESET WRITE POINTER command processing

ZONE COUNT field	ALL bit	Description
zero or one	zero	If the specified zone has a Zone Condition of: a) EMPTY, then the device server shall make no changes in the Zone Condition and not return an error; and b) IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or FULL, then the device server shall perform a reset write pointer zone operation (see 4.5.3.2.5) on the specified zone.
two or more	zero	For each zone in the specified zone range that has a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or FULL, the device server shall perform a reset write pointer operation.
zero	one	For each zone that has: a) a Restrict Write Pointer Reset realm attribute (see 4.4.2) that is false and b) a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or FULL, the device server shall perform a reset write pointer operation.
non-zero	one	See 5.1.2.

5.10 SEQUENTIALIZE ZONE command

The SEQUENTIALIZE ZONE command (see table 47) requests the device server to perform sequentialize zone operations (see 4.5.3.2.6).

Table 47 – SEQUENTIALIZE ZONE command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (94h)							
1	Reserved			SERVICE ACTION (10h)				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10								
11	Reserved							
12	(MSB)							
13	ZONE COUNT							(LSB)
14	Reserved							ALL
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 47 for the SEQUENTIALIZE ZONE command.

The ZONE ID field, ZONE COUNT field, and ALL bit are defined in 5.1.2.

The device server shall terminated 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, if:

a) the ZONE ID field does not specify a sequential write preferred zone (see 4.5.3.3); or
b) the specified zone range, if any, includes a zone that is not a sequential write preferred zone.

The CONTROL byte is defined in SAM-6.

If the SEQUENTIALIZE ZONE command is not terminated with CHECK CONDITION status (see 5.1.2), then the device server processes the command as described in table 48.

Table 48 – SEQUENTIALIZE ZONE command processing

ZONE COUNT field	ALL bit	Description
zero or one	zero	If the specified zone has a Zone Condition of: a) EMPTY, then the device server shall make no changes in the Zone Condition and not return an error; and b) IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or FULL, then the device server shall perform a sequentialize zone operation (see 4.5.3.2.6) on the specified zone.
two or more	zero	For each zone in the specified zone range that has a Zone Condition of CLOSED, the device server shall perform a sequentialize zone operation.
zero	one	For each sequential write preferred zone (see 4.5.3.3) that has a Zone Condition of IMPLICITLY OPENED, EXPLICITLY OPENED, CLOSED, or FULL, the device server shall perform a sequentialize zone operation.
non-zero	one	See 5.1.2.

5.11 ZONE ACTIVATE command

5.11.1 ZONE ACTIVATE command overview

The ZONE ACTIVATE command (see table 49) requests that the device server perform a zone activation operation (see 4.5.3.2.7) that:

a) changes which zones are active zones and which zones are inactive zones; and
b) transfers the resulting parameter data.

The device server shall process each ZONE ACTIVATE command as if it has an ORDERED task attribute (see SAM-6).

Table 49 – ZONE ACTIVATE command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (95h)							
1	ALL	Reserved		SERVICE ACTION (08h)				
2	(MSB)							
•••	ZONE ID							
9								(LSB)
10	(MSB)							
11	NUMBER OF ZONES							(LSB)
12	(MSB)							
13	ALLOCATION LENGTH							(LSB)
14	OTHER ZONE DOMAIN ID							
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 49 for the ZONE ACTIVATE command.

An ALL bit set to:

a) zero specifies that the zone activation operation is not requested to process all zones; and
b) one specifies that the zone activation operation is requested to process all zones.

If the ALL bit is set to zero, the ZONE ID field specifies the lowest LBA of the write pointer zone (see 4.5.3) for which performing a zone activation operation is requested.

If the ALL bit is set to one, the ZONE ID field is ignored.

If the ALL bit is set to zero, the NUMBER OF ZONES field specifies the number of write pointer zones starting with the LBA specified in the ZONE ID field for which performing a zone activation operation is requested.

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, if the ALL bit is set to zero and:

a) the NUMBER OF ZONES field is set to zero;
b) one or more LBAs in the specified zone range is:
   A) not contained in a write pointer zone; or

B) associated with a Restrict Zone Activate realm attribute (see 4.4.3) that is true; or

c) the AAORB bit (see 6.5.2) is set to one and the specified zone range does not:
   A) start on a realm boundary; and
   B) end on a realm boundary.

If the ALL bit is set to one, the NUMBER OF ZONES field is ignored.

The ALLOCATION LENGTH field is defined in SPC-6. If the allocation length is less than 64, 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 OTHER ZONE DOMAIN ID field specifies a second zone domain, if any, for which performing a zone activation operation is requested (see 5.11.2). In some cases, the device server is able to determine all necessary zone domain information without a specified second domain. If table 50 does not describe the OTHER ZONE DOMAIN ID field as ignored and the OTHER ZONE DOMAIN ID field contains a zone domain ID (see 4.2.4.2) that is not supported by the device server, 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 CONTROL byte is defined in SAM-6.

The parameter data returned by a ZONE ACTIVATE command is defined in 5.11.3.

The device server shall process the ZONE ACTIVATE command by:

1) identifying the candidate zones to activate and the candidate zones to deactivate (see 5.11.2); and
2) if no errors are detected in step 1), then performing a zone activation operation (see 4.5.3.2.7).

5.11.2 Identifying the candidate zones to activate and the candidate zones to deactivate

This subclause (i.e., 5.11.2) defines how a ZONE ACTIVATE command (see 5.11) or a ZONE QUERY command (see 5.12) identifies:

a) the candidate zones to active; and
b) the candidate zones to deactivate.

The candidate zones to activate and the candidate zones to deactivate that are inputs to the zone activation operation (see 4.5.3.2.7).

The command fields (see 5.11.1) used in this subclause (i.e., 5.11.2) are:

a) the ALL bit;
b) the ZONE ID field;
c) the NUMBER OF ZONES field; and
d) the OTHER ZONE DOMAIN ID field.

This subclause (i.e., 5.11.2) ignores some command fields if other command fields specify equivalent information (e.g., if the ALL bit is set to one, the NUMBER OF ZONES field is ignored).

If the ALL bit is set to one, then:

a) the candidate zones to activate are all zones in the zone domain specified by the OTHER ZONE DOMAIN ID field that are associated with a Restrict Zone Activate realm attribute (see 4.4.3) that is false; and
b) the candidate zones to deactivate are zones that are:
   A) in any zone domain except the zone domain specified by the OTHER ZONE DOMAIN ID field; and
   B) associated with a Restrict Zone Activate realm attribute that is false.

If the ALL bit is set to zero, then:

a) the specified zone range is the set of zones starting with the value of the ZONE ID field and extending for the number of zones specified by the NUMBER OF ZONES field; and
b) table 50 defines how the specified zone range is used to identify the candidate zones to activate and the candidate zones to deactivate.

Table 50 – Selecting candidate zones to activate and deactivate with ALL bit set to zero (part 1 of 2)

<table>
<tr><th>Zone Conditions in the specified zone range</th><th>Activation aligned on realm boundaries [a]</th><th>Candidate zones to activate</th><th>Candidate zones to deactivate</th></tr>
<tr><td rowspan="2">All specified zones are inactive zones</td><td>No</td><td rowspan="2">The zone domain to activate is the zone domain that contains the specified zone range.
The candidate zones to activate are the zones in the specified zone range.
The OTHER ZONE DOMAIN ID field is ignored.</td><td>The zone domain and the candidate zones to deactivate are determined by the device server using the specified zone range. [b]
The device server shall select zones to deactivate that have a Zone Condition of EMPTY or INACTIVE.</td></tr>
<tr><td>Yes</td><td>The zone domain and candidate zones with a Zone Condition of EMPTY to deactivate are determined by the device server using the realm and zone domain of the specified zone range.</td></tr>
<tr><td rowspan="2">All specified zones are active zones</td><td>No</td><td>The zone domain to activate is specified by the OTHER ZONE DOMAIN ID field.
The candidate zones to activate in that zone domain are determined by the device server using the zone range that is specified to be deactivated. [c]
The device server shall not select any zone to activate if that zone has a Zone Condition that is not INACTIVE.</td><td rowspan="2">The zone domain to deactivate is the zone domain of the specified zone range.
The candidate zones to deactivate are the zones in the specified zone range.</td></tr>
<tr><td>Yes</td><td>The zone domain to activate is specified by the OTHER ZONE DOMAIN ID field.
The candidate zones to activate with a Zone Condition of INACTIVE are determined by the device server using the realm and zone domain of the zone range that is specified to be deactivated.</td></tr>
</table>

[a] Is the AAORB bit (see 6.5.2) set to one?
[b] The candidate zones to deactivate should include the smallest possible number of zones. There may be zero candidate zones to deactivate.
[c] The candidate zones to activate should include the largest possible number of zones.

Table 50 – Selecting candidate zones to activate and deactivate with ALL bit set to zero (part 2 of 2)

Zone Conditions in the specified zone range	Activation aligned on realm boundaries [a]	Candidate zones to activate	Candidate zones to deactivate
Some specified zones are active zones and some specified zones are inactive zones	n/a	The candidate zones to activate are the specified zone range so that an error is able to be reported as described in 4.5.3.2.7.2.	

[a] Is the AAORB bit (see 6.5.2) set to one?
[b] The candidate zones to deactivate should include the smallest possible number of zones. There may be zero candidate zones to deactivate.
[c] The candidate zones to activate should include the largest possible number of zones.

5.11.3 ZONE ACTIVATE parameter data and ZONE QUERY parameter data

5.11.3.1 ZONE ACTIVATE parameter data and ZONE QUERY parameter data overview

The ZONE ACTIVATE parameter data and ZONE QUERY parameter data (see table 51) is returned whenever the command completes with GOOD status.

Table 51 – ZONE ACTIVATE parameter data and ZONE QUERY parameter data (part 1 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
•••	ZONE ACTIVATION RESULTS LENGTH (n–63)							
3								(LSB)
4	(MSB)							
•••	ZONE ACTIVATION RESULTS RETURNED LENGTH							
7								(LSB)
8	NZ_VALID	ZIWUP_VALID	Reserved					ACTIVATED
9	Reserved	SECURITY	MULT DOMN	RLM RSTCT	MULT ZTYP	RLM ALIGN	NOT EMPTY	NOT INACT
10	OTHER ZONE DOMAIN ID							
11	Reserved							ALL
12	Reserved							
13	Reserved							
•••								
15								

Table 51 – ZONE ACTIVATE parameter data and ZONE QUERY parameter data (part 2 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
16	(MSB)							
•••				NUMBER OF ZONES				
19								(LSB)
20								
•••				Reserved				
23								
24	(MSB)							
•••				ZONE ID WITH UNMET PREREQUISITE				
31								(LSB)
32								
•••				Reserved				
63								
Zone activation descriptors list								
64								
•••				Zone activation descriptor [first] (see 5.11.3.2)				
95								
				⋮				
n–31								
•••				Zone activation descriptor [last] (see 5.11.3.2)				
n								

The ZONE ACTIVATION RESULTS LENGTH field indicates the length in bytes of the zone activation descriptors list. The contents of the ZONE ACTIVATION RESULTS LENGTH field shall not be altered based on the allocation length (see SPC-6).

The ZONE ACTIVATION RESULTS RETURNED LENGTH field indicates the length in bytes of the part of the zone activation descriptors list that are returned for the application client to process. If the allocation length (see SPC-6) is not sufficient to contain all of the contents of the zone activation descriptors list, then the ZONE ACTIVATION RESULTS RETURNED LENGTH field shall be altered to indicate the truncation of the zone activation descriptors list.

An NZ_VALID bit set to zero indicates that the NUMBER OF ZONES field does not contain valid information (i.e., the ALL bit is set to one). An NZ_VALID bit set to one indicates that the NUMBER OF ZONES field contains valid information (i.e., the ALL bit is set to zero).

A ZIWUP_VALID bit set to zero indicates that the ZONE ID WITH UNMET PREREQUISITE field does not contain valid information. A ZIWUP_VALID bit set to one indicates that the ZONE ID WITH UNMET PREREQUISITE field contains valid information.

An ACTIVATED bit set to one indicates that the specified change activations operation was performed (see 4.5.3.2.7.3). An ACTIVATED bit set to zero indicates that no zones were changed. The ACTIVATED bit also indicates what is described by each zone activation descriptor (see 5.11.3.2)

The SECURITY (i.e., security prerequisites) bit, multiple domains (MULT DOMN) bit, realm restrictions (RLM RSTCT) bit, multiple zone types (MULT ZTYP) bit, realm alignment (RLM ALIGN) bit, NOT EMPTY bit, and not inactive (NOT INACT) bit indicate unmet prerequisites as described in 4.5.3.2.7.2.

The OTHER ZONE DOMAIN ID field contains a copy of the OTHER ZONE DOMAIN ID field in the ZONE ACTIVATE command (see 5.11.1) or the ZONE QUERY command (see 5.12).

The ALL bit contains a copy of the ALL bit in the ZONE ACTIVATE command or the ZONE QUERY command.

The NUMBER OF ZONES field contains a copy of the NUMBER OF ZONES field in the ZONE ACTIVATE command or the ZONE QUERY command. The NZ_VALID bit indicates whether the contents of the NUMBER OF ZONES field are valid.

If the ZIWUP_VALID bit is set to one, then the ZONE ID WITH UNMET PREREQUISITE field indicates the zone ID of the lowest numbered zone associated with an unmet prerequisite.

Each zone activation descriptor (see 5.11.3.2) contains information about a zone range associated with an activation or deactivation. Zone activation descriptors are not returned if the verify activations operation (see 4.5.3.2.7.2) detects one or more unmet prerequisites.

5.11.3.2 Zone activation descriptors

If the ACTIVATED bit (see 5.11.3.1) is set to:

a) one, then each zone activation descriptor (see table 52) indicates one zone range that has been activated or deactivated by the ZONE ACTIVATE command returning that descriptor; and
b) zero, then each zone activation descriptor indicates one zone range that may be activated or deactivated by a subsequent ZONE ACTIVATE command with command inputs that are identical to those for the command that is returning that descriptor.

If the ALL bit (see 5.11.3.1) is set to zero, then zone activation descriptors are returned by a ZONE ACTIVATE command or a ZONE QUERY command (see 5.12) that:

a) does not detect unmet prerequisites; or
b) detects only unmet prerequisites with the result that one or more of the following bits (see 5.11.3.1) are set to one:
   A) the SECURITY bit;
   B) the NOT EMPTY bit; or
   C) the NOT INACT bit.

Zone activation descriptors are not returned by a ZONE ACTIVATE command or a ZONE QUERY command that:

a) has the ALL bit set to one; or
b) detects any unmet prerequisites with the result that one or more of the following bits (see 5.11.3.1) are set to one:
   A) the MULT DOMN bit;
   B) the RLM RSTCT bit;
   C) the MULT ZTYP bit; or
   D) the RLM ALIGN bit.

Table 52 shows the format of a zone activation descriptor.

Table 52 – Zone activation descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved				ZONE TYPE			
1	ZONE CONDITION				Reserved			
2	ZONE DOMAIN ID							
3	Reserved							
•••								
7								
8	(MSB)							
•••	ZONE RANGE SIZE							
15								(LSB)
16	(MSB)							
•••	STARTING ZONE LOCATOR							
23								(LSB)
24	Reserved							
•••								
31								

The ZONE TYPE field (see table 10) indicates the Zone Type (see 4.3.2) for all zones described by this zone activation descriptor.

The ZONE CONDITION field (see table 12) indicates the Zone Condition (see 4.3.3) for all zones described by this zone activation descriptor.

The ZONE DOMAIN ID field indicates the zone domain ID (see 4.2.4.2) for all zones described by this zone activation descriptor.

The ZONE RANGE SIZE field indicates the number of zones described by this zone activation descriptor.

The STARTING ZONE LOCATOR field indicates the lowest LBA of the first zone described by this zone activation descriptor.

5.12 ZONE QUERY command

The ZONE QUERY command (see table 53) requests that the device server process a zone activation operation (see 4.5.3.2.7) that:

a) does not change which zones are active zones and which zones are inactive zones; and
b) transfers the resulting parameter data.

Table 53 – ZONE QUERY command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (95h)							
1	ALL	Reserved		SERVICE ACTION (09h)				
2	(MSB)							
•••		ZONE ID						
9								(LSB)
10	(MSB)							
11		NUMBER OF ZONES						(LSB)
12	(MSB)							
13		ALLOCATION LENGTH						(LSB)
14	OTHER ZONE DOMAIN ID							
15	CONTROL							

The OPERATION CODE field and the SERVICE ACTION field are defined in SPC-6 and shall be set to the values shown in table 53 for the ZONE QUERY command.

The ALL bit, ZONE ID field, NUMBER OF ZONES field, and OTHER ZONE DOMAIN ID field are defined in 5.11.1.

The ALLOCATION LENGTH field is defined in SPC-6. If the allocation length is less than 64, 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 CONTROL byte is defined in SAM-6.

The parameter data returned by a ZONE QUERY command is defined in 5.11.3.

The device server shall process the ZONE QUERY command by:

1) identifying the candidate zones to activate and the candidate zones to deactivate (see 5.11.2);
2) prepare the parameter data defined in 5.11.3; and
3) perform a verify activations operation (see 4.5.3.2.7.2).

If the device server processes zone activation operation for the ZONE QUERY command, then the device server shall not terminate the command with CHECK CONDITION status.

6 Parameters for zoned block devices

6.1 Parameters for zoned block devices overview

Table 54 shows the parameters for zoned block devices defined in clause 6 and a reference to the subclause where each parameter type is defined.

Table 54 – Parameters for zoned block devices

Parameter type	Reference
Diagnostic parameters	6.2
Log parameters	6.3
Mode parameters	6.4
Vital product data (VPD) parameters	6.5

6.2 Diagnostic parameters

The diagnostic pages and their corresponding page codes for zoned block devices are defined in table 55.

Table 55 – Diagnostic page codes for host managed zoned block devices

Diagnostic page	Page code	Reference
Diagnostic pages assigned by SPC-6	30h to 3Fh	SPC-6
Direct access device diagnostic pages	40h to 7Fh	SBC-5
SCSI enclosure services diagnostic pages	01h to 2Fh	SES-4
Vendor specific diagnostic pages	80h to FFh	