Working Draft

Project T10/1611-D

Information technology -
SCSI Stream Commands - 3 (SSC-3)

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

T10 Technical Editor: David Peterson
                         Computer Technology Corporation
                         6000 Nathan Lane North
                         Plymouth, MN 55442
                         USA

                         Telephone: 1-763-268-6139
                         Email: dap@cnt.com

Printed Thursday, February 10, 2005 1:39 PM

Points of Contact:

Rev 00 Changes:

- converted from SSC-2 rev 09
- changed references from SAM-2 to SAM-3
- 03-289r3: Only If Reserved Proposal
- removed asynchronous event reporting
- added SSC sense data descriptor from SPC-3 r17 (subclause 4.5.2.6 Stream commands sense data descriptor)

Rev 01 Changes:

- Completed hanging sentence in subclause 8.3.3 text for SWP bit and removed the redundent next paragraph.
- 04-123r0: Exclude REPORT LUNS from resetting TEST bit
- 04-119r0: DTD Status log page
- modified text for ASOCWP in 8.3.3
- modified text for PERSWP in 8.3.3
- modified text for PRMWP in 8.3.3

Rev 01a Changes:

- added text for SWP in 8.3.3

Rev 01b Changes:

04-211r1: WORM support for streaming devices
04-104r2: Add report to REPORT DENSITY SUPPORT command
04-178r2: Add device type specific VPD page
- broke READ POSITION command into subclauses

Rev 01c Changes:

04-283r1: Reservation model cleanup
05-014r0: Fix REPORT DENSITY SUPPORT data
05-037r0: Synchronize Mode Page Table with SPC-3
04-335r2: WORM Tamper Read Enable
05-373r2: Model subclause for WORM
05-066r0: Obsolete GAP SIZE field
05-067r0: Obsolete RSMK and setmarks

Draft

**American National Standards
for Information Systems -**

SCSI Stream Commands - 3 (SSC-3)

Secretariat
National Committee for Information Technology Standards

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This standard specifies the device model and functional requirements for the SCSI sequential-access stream device type. This standards permits the SCSI sequential-access stream device type to attach to computers and provides the definitions for their use.

This standard does not contain material related to any service delivery subsystem which is used to transport the commands, command parameter data, command response data, and status specified in this standard.

Draft

American National Standard

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

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

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

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

Draft

Contents

Tables

Figures

Foreword

This foreword is not part of American National Standard INCITS.***:200x.

This standard specifies the external behavior of a device server that defines itself as a sequential-access device in the DEVICE TYPE field of the INQUIRY command response data. This device type is known as a stream device. This standard conforms to the SCSI Architecture Model - 3 (T10/1561-D) standard.

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

These Bulletins, while reflecting the opinion of the Technical Committee that developed the standard, are intended solely as supplementary information to other users of the standard. This standard, ANSI INCITS.***:200x, as approved through the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of these Technical Information Bulletins.

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

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

or

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

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 Institute, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

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

<<Insert INCITS member list>>

The INCITS Technical Committee T10 on Lower Level Interfaces, which reviewed this standard, had the following members:

<<Insert T10 member list>>

Introduction

The SCSI Stream Commands - 3 (SSC-3) standard is divided into eight clauses:

Clause 1 is the scope.

Clause 2 enumerates the normative references that apply to this standard.

Clause 3 describes the definitions, symbols, and abbreviations used in this standard.

Clause 4 describes an overview and model of the sequential-access type device.

Clause 5 describes the explicit address command set for sequential-access type devices.

Clause 6 describes the implicit address command set for sequential-access type devices.

Clause 7 describes the common command set for sequential-access type devices.

Clause 8 describes the parameters for sequential-access type devices.

The annexes provide information to assist with implementation of this standard.

American National Standard **INCITS.***:200x**

**American National Standard for Information Systems -
Information Technology -
SCSI Stream Commands - 3 (SSC-3)**

1 Scope

This standard defines the command set extensions to facilitate operation of the sequential-access device type member of the SCSI stream device class. The clauses of this standard, implemented in conjunction with the applicable clauses of the SCSI Primary Commands - 3 standard, fully specify the standard command set for the sequential-access device type member of the SCSI stream device class.

The objectives of this standard are to provide the following:

    a) permit an application client to communicate over a SCSI service delivery subsystem, with a logical unit that declares itself to be a sequential-access device in the device type field of the INQUIRY command response data;
    b) define commands unique to the sequential-access device type; and
    c) define commands to manage the operation of the sequential-access device type.

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

Architecure Model	
	Device-Type Specific Command Sets
	Shared Command Set (for all device types)
	SCSI Protocols
	Interconnects
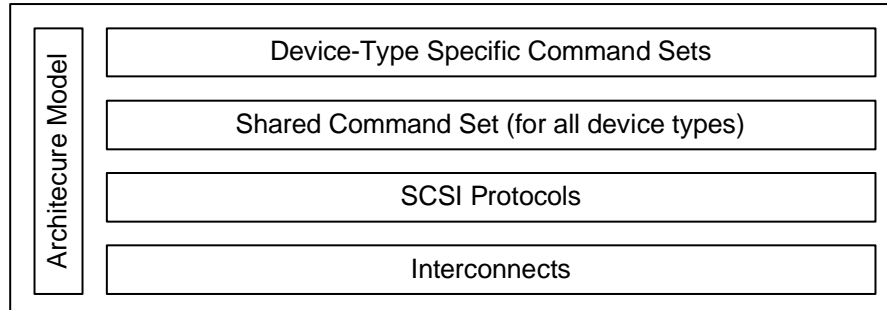
Figure 1 — SCSI document relationships

The roadmap in figure 1 is intended to show the general applicability of the documents to one another. The figure is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture. It specifies the applicability of a standard to the implementation of a given SCSI protocol.

At the time this standard was generated, examples of the SCSI general structure included:

Physical Interconnects:
Fibre Channel Arbitrated Loop	FC-AL	[ISO/IEC 14165-121]
		[ANSI X3.272:1996]
Fibre Channel Physical and Signalling Interface	FC-PH	[ISO/IEC 14165-111]
		[ANSI X3.230:1994]
Fibre Channel Physical Amendment 1		[ANSI X3.230/AM1:1996]
Fibre Channel 3rd Generation Physical Interface	FC-PH-3	[ISO/IEC 14165-113]
		[ANSI X3.303-1998]
Fibre Channel Framing and Signaling Interface	FC-FS	[T11/1331-D]
High Performance Serial Bus		[ANSI IEEE 1394:1995]
SCSI Parallel Interface - 2	SPI-2	[ISO/IEC 14776-112]
		[ANSI X3.302:1999]
SCSI Parallel Interface - 3	SPI-3	[ISO/IEC 14776-113]
		[ANSI NCITS.336:200x]
SCSI Parallel Interface - 4	SPI-4	[T10/1365-D]
Serial Storage Architecture Physical Layer 1	SSA-PH	[ANSI X3.293:1996]
Serial Storage Architecture Physical Layer 2	SSA-PH-2	[ANSI NCITS.307:1998]

Transport Protocols:
Serial Storage Architecture Transport Layer 1	SSA-TL-1	[ANSI X3.295:1996]
Serial Storage Architecture Transport Layer 2	SSA-TL-2	[ANSI NCITS.308:1998]
SCSI-3 Fibre Channel Protocol	FCP	[ISO/IEC 14776-221]
		[ANSI X3.269:1996]
SCSI-3 Fibre Channel Protocol - 2	FCP-2	[ISO/IEC 14776-222]
		[T10/1144-D]
Serial Bus Protocol - 2	SBP-2	[ISO/IEC 14776-232]
		[ANSI NCITS.325:1999]
Serial Storage Architecture SCSI-3 Protocol	SSA-S3P	[ANSI NCITS.309:1998]
SCSI on Scheduled Transfer	SST	[T10/1380-D]
SCSI RDMA Protocol	SRP	[T10/1415-D]

Shared Command Sets:
SCSI-3 Primary Commands	SPC	[ANSI X3.301:1997]
SCSI Primary Commands - 2	SPC-2	[ISO/IEC 14776-312]
		[T10/1236-D]
SCSI Primary Commands - 3	SPC-3	[T10/1416-D]

Device-Type Specific Command Sets:
SCSI-3 Block Commands	SBC	[ISO/IEC 14776-321]
		[ANSI NCITS.306:1998]
SCSI Block Commands - 2	SBC-2	[T10/1417-D]
SCSI-3 Stream Commands	SSC	[ISO/IEC 14776-331]
		[ANSI NCITS.335:200x]
SCSI Stream Commands - 2	SSC-2	[T10/1434-D]
SCSI-3 Medium Changer Commands	SMC	[ISO/IEC 14776-351]
		[ANSI NCITS.314:1998]
SCSI Medium Changer Commands - 2	SMC-2	[T10/1383-D]
SCSI-3 Multimedia Command Set	MMC	[ANSI X3.304:1997]
SCSI Multimedia Command Set - 2	MMC-2	[ISO/IEC 14776-362]
		[ANSI NCITS.333:200x]
SCSI Multimedia Command Set - 3	MMC-3	[T10/1363-D]

SCSI-3 Controller Commands	SCC	[ISO/IEC 14776-341]
		[ANSI X3.276:1997]
SCSI Controller Commands - 2	SCC-2	[ISO/IEC 14776-342]
		[ANSI NCITS.318:1998]
SCSI Reduced Block Commands	RBC	[ANSI NCITS.330:200x]
SCSI Reduced MultiMedia Commands	RMC	[T10/1364-D]
SCSI-3 Enclosure Services Commands	SES	[ANSI NCITS.305:1998]
SCSI Specification for Optical Card Reader/Writer	OCRW	[ISO/IEC 14776-381]
Object-based Storage Devices Commands	OSD	[T10/1355-D]

Architecture Model:

SCSI-3 Architecture Model	SAM	[ISO/IEC 14776-411]
		[ANSI X3.270:1996]
SCSI Architecture Model - 2	SAM-2	[ISO/IEC 14776-412]
		[T10/1157-D]

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

2 Normative references

2.1 Normative references

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

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

2.2 Approved references

ISO/IEC 14776-411, *SCSI-3 Architecture Model standard*
ISO/IEC 14776-312, *SCSI Primary Commands - 2 standard*

2.3 References under development

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

ISO/IEC 14776-313, *SCSI Primary Commands - 3 standard*
ISO/IEC 14776-412, *SCSI Architecture Model - 2 standard*
ISO/IEC 14776-352, *SCSI Media Changer Commands - 2 standard*

3 Definitions, acronyms, keywords, and conventions

3.1 Definitions

3.1.1 additional sense code: See SPC-3.

3.1.2 application client: An object that is the source of SCSI commands and task management requests. Further definition of an application client may be found in SAM-3.

3.1.3 auxiliary memory: A memory component that is accessible to the device server. This memory is usually non-volatile and independent of the main function of the device server.

3.1.4 BOx: Either beginning-of-medium (see 3.1.5) or beginning-of-partition (see 3.1.6).

3.1.5 beginning-of-medium (BOM): The extreme position along the medium in the direction away from the supply reel that is accessible by the device. This position may not coincide with a beginning-of-partition position.

3.1.6 beginning-of-partition (BOP): The position at the beginning of the permissible recording region of a partition. This position may not coincide with a beginning-of-medium position.

3.1.7 block address mode: The mode of operation that the logical unit is currently supporting (see 4.2.14). The block address mode is either the explicit address mode (see 3.1.19) or the implicit address mode (see 3.1.28).

3.1.8 buffered mode: A mode of data transfer in write operations that facilitates tape streaming. Buffered mode is specified by a non-zero value (1h or 2h) in the BUFFER MODE field in the mode parameter header (see 8.3). Buffered mode is the opposite of unbuffered mode (see 3.1.70).

3.1.9 byte: An 8-bit construct.

3.1.10 command: A request describing a unit of work to be performed by a device server. A detailed definition of a command may be found in SAM-3.

3.1.11 command descriptor block (CDB): The structure used to communicate commands from an application client to a device server. A command descriptor block may have a fixed length of up to 16 bytes or a variable length of between 12 and 260 bytes.

3.1.12 common command: A command that is contained in both the explicit and implicit address command sets.

3.1.13 device server: An object within a logical unit that processes SCSI tasks according to the rules of task management. A detailed definition of a device server may be found in SAM-3.

3.1.14 device type: The type of device (or device model) implemented by the device server.

3.1.15 early-warning (EW): A physical mark or device computed position near but logically before the end-of-partition, independent of physical direction (see 4.2.3).

3.1.16 end-of-data (EOD): A recorded indication that no valid logical objects are recorded between this position and end-of-partition. End-of-data is denoted in a format-specific manner (see 4.2.5.1).

3.1.17 end-of-medium (EOM): The extreme position along the medium in the direction away from the take-up reel that is accessible by the device. This position may not coincide with a end-of-partition position.

3.1.18 end-of-partition (EOP): The position at the end of the permissible recording region of a partition.

3.1.19 explicit address mode: The mode of operation in which the logical unit is supporting the explicit address command set (see 3.1.20).

3.1.20 explicit address command set: The command set in which read and writes contain positioning information.

3.1.21 explicit command: A command contained only in the explicit address command set (see table 12).

3.1.22 field: A group of one or more contiguous bits, a part of a larger structure such as a CDB (see 3.1.11) or sense data (see 3.1.59).

3.1.23 filemark: A special recorded logical object within a partition, not containing user data, that provides a segmentation scheme for the contents of a partition.

3.1.24 fixed-block transfer: A read or write type command with the FIXED bit set to one.

3.1.25 format label: A vendor-specific series of logical objects that contain information used to identify the volume or data set.

3.1.26 gap: A non-data object recorded on the medium. Gaps may be recorded between logical objects. The format and method of recording a gap may vary.

3.1.27 generic command: An explicit command (see 3.1.21) that is not a read type or write type command.

3.1.28 implicit address mode: The mode of operation in which the logical unit is supporting the implicit address command set (see 3.1.29).

3.1.29 implicit address command set: The command set in which read and writes do not contain positioning information, and positioning is implied relative to the current position.

3.1.30 implicit command: A command contained only in the implicit address command set (see table 19).

3.1.31 INFORMATION field: A command-specific field in the sense data (see SPC-3).

3.1.32 I_T nexus: A nexus between a SCSI initiator port and a SCSI target port (see SAM-3).

3.1.33 I_T nexus loss: A condition resulting from the events defined by SAM-3 in which the SCSI device performs the operations described in SAM-3 and this standard.

3.1.34 I_T nexus loss event: A SCSI transport protocol specific event that triggers I_T nexus loss as described in SAM-3.

3.1.35 logical block: A logical object that is a unit of data supplied or requested by an application client.

3.1.36 logical file: Zero or more logical blocks starting immediately after BOP or a filemark (see 4.2.6).

3.1.37 logical file identifier: A unique identifier, within a partition, for a logical file (see 4.2.6.2).

3.1.38 logical identifier: A logical object identifier or logical file identifier.

3.1.39 logical object: A logical block or a filemark (see 4.2.5).

3.1.40 logical object identifier: A unique identifier, within a partition, for a logical object (see 4.2.5.2).

3.1.41 logical unit reset: A logical unit action in response to a logical unit reset event in which the logical unit performs the operations described in SAM-3.

3.1.42 logical unit reset event: An event that triggers a logical unit reset from a logical unit as described in SAM-3.

3.1.43 medium auxiliary memory (MAM): An auxiliary memory residing on a medium, for example, a tape cartridge.

3.1.44 nexus: A relationship between two SCSI devices, and the SCSI initiator port and SCSI target port objects within those devices (see SAM-3).

3.1.45 one: The logical true condition of a variable.

3.1.46 overlength: The incorrect-length condition that exists after processing a read command when the length of the actual logical block read exceeds the requested transfer length in the command descriptor block or the mode header block size field, whichever is appropriate.

3.1.47 page: Several commands use regular parameter structures that are referred to as pages. These pages are identified with a value known as a page code.

3.1.48 partition: The entire usable region for recording and reading in a volume or in a portion of a volume, defined in a vendor-specific or format-specific manner (see 4.2.4).

3.1.49 principal density code: The principal density code is a density code selected by the device server. The logical unit specifies the principal density code by reporting a DEFLT bit of one in the density support data block descriptor for supported densities in response to the REPORT DENSITY SUPPORT command (see 7.7). The selection of the principal density code is vendor-specific.

3.1.50 SCSI device: A device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol (see SAM-3).

3.1.51 SCSI domain: The interconnection of two or more SCSI devices and a service delivery subsystem. A detailed definition of a SCSI domain may be found in SAM-3.

3.1.52 SCSI initiator device: A SCSI device containing application clients and SCSI initiator ports that originates device service and task management requests to be process by a SCSI target device and receives device service and task management responses from the SCSI target devices. When used this term refers to SCSI initiator devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI initiator port.

3.1.53 SCSI initiator port: A SCSI initiator device object that acts as the connection between application clients and the service delivery subsystem through which requests, indications, responses, and confirmations are routed. In all cases when this term is used it refers to a SCSI initiator port or a SCSI target/initiator port operating as a SCSI initiator port.

3.1.54 SCSI port: A SCSI device resident object that connects the application client, device server or task manager to the service delivery subsystem through which requests and responses are routed. SCSI port is synonmous with port. A SCSI port is one of: a SCSI initiator port, a SCSI target port, or a SCSI target/initiator port.

3.1.55 SCSI target device: A SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing and sends device service and task management responses to SCSI initiator devices. When used this term refers to SCSI target devices or SCSI target/initiator devices that are using the SCSI target/initiator port as a SCSI target port.

3.1.56 SCSI target port: A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and the service delivery subsystem through which indications and responses are routed. When this term is used it refers to a SCSI target port or a SCSI target/initiator port operating as a SCSI target port.

3.1.57 SCSI target/initiator device: A SCSI device that has all the characteristics of a SCSI target device and a SCSI initiator device.

3.1.58 SCSI target/initiator port: A SCSI device resident object that has all the characteristics of a SCSI target port and a SCSI initiator port.

3.1.59 sense data: See SPC-3.

3.1.60 sense key: See SPC-3.

3.1.61 service delivery subsystem: That part of an SCSI I/O system that transmits service requests to a logical unit or SCSI target device and returns logical unit or SCSI target device responses to a SCSI initiator device. A detailed definition of a service delivery subsystem may be found in SAM-3.

3.1.62 spacing: The act of positioning the medium on a sequential-access device while processing a SPACE command.

3.1.63 status: One byte of response information sent from a device server to an application client upon completion of each command. A detailed definition of status may be found in SAM-3.

3.1.64 synchronize operation: The process of writing buffered logical objects to the medium (see 4.2.8).

3.1.65 tagged write sequence: One or more WRITE(16), WRITE FILEMARKS(16), or ERASE(16) commands delineated by the FCS and LCS bits (see 5.6, 5.7, and 5.2).

3.1.66 tape: The medium on which data is recorded. The medium is normally a long thin medium that is spooled onto one or two reels, possibly within a cassette or cartridge.

3.1.67 TapeAlert: A device server cpapbility that provides detailed device diagnostic information using a standard interface.

3.1.68 track: A contiguous line on the medium consisting of a pattern of recorded signals written by one write component.

3.1.69 track group: A set of tracks that are recorded at the same time.

3.1.70 unbuffered mode: The mode of operation where write data is written directly to the medium without being buffered. Unbuffered mode is specified by a zero value (0h) in the BUFFER MODE field in the mode parameter header (see 8.3). Unbuffered mode is the opposite of buffered mode (see 3.1.8).

3.1.71 underlength: The incorrect-length condition that exists after processing a read command when the requested transfer length in the command descriptor block or the mode header block size field, whichever is appropriate, exceeds the length of the actual logical block read.

3.1.72 variable-block transfer: A read or write type command with the FIXED bit set to zero.

3.1.73 volume: A recording medium together with its physical carrier.

3.1.74 zero: The logical false condition of a variable.

3.2 Acronyms

ALDC	Adaptive Lossless Data Compression: ISO/IEC 15200:1996
BOM	beginning-of-medium
BOP	beginning-of-partition
CDB	command descriptor block
DCLZ	Data Compression according to Lempel and Ziv: ISO/IEC 11558:1992
ECC	error correction code
ECMA	European Computer Manufacturers Association
EOD	end-of-data
EOM	end-of-medium
EOP	end-of-partition
EW	early-warning
IDRC	Improved Data Recording Capability
I/O	input-output
ID	identifier
LSB	least significant bit
M	mandatory
MAM	medium auxiliary memory
MSB	most significant bit
NA	not applicable
O	optional
Rsvd	reserved
INCITS	InterNational Committee for Information Technology Standards
SAM-3	SCSI Architecture Model - 3
SBC	SCSI-3 Block Commands
SCSI	Small Computer System Interface
SCSI-3	Small Computer System Interface - 3
SMC-2	SCSI Media Changer Commands - 2
SPC-2	SCSI Primary Commands - 2
SPC-3	SCSI Primary Commands - 3
SSC-2	SCSI Stream Commands - 2 (this standard)

3.3 Keywords

3.3.1 expected: A keyword used to describe the behavior of the hardware or software in the design models assumed by this standard. Other hardware and software design models may also be implemented.

3.3.2 invalid: A keyword used to describe an illegal or unsupported field or code value. Receipt of an invalid field or code value shall be reported as an error.

3.3.3 ignored: A keyword used to describe an unused field or code value. The contents or value of an ignored field or code value shall not be examined by the receiving SCSI device and may be set to any value by the transmitting SCSI device.

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

3.3.5 may: A keyword that specifies flexibility of choice with no implied preference (equivalent to "may or may not").

3.3.6 may not: A keyword that specifies flexibility of choice with no implied preference (equivalent to "may or may not").

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

3.3.8 optional: A keyword that describes features that are not required to be implemented by this standard. However, if any optional feature defined by this standard is implemented, then it shall be implemented as defined in this standard.

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

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

3.3.11 should: A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase "it is strongly recommended".

3.3.12 vendor-specific: Items (e.g., fields, code values, etc.) that are not defined by this standard and may be vendor defined.

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 clause 3.1 or in the text where they first appear. Names of commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE). Lowercase is used for words having the normal English meaning.

The names of fields are in small uppercase (e.g., ALLOCATION LENGTH). When a field name is a concatenation of acronyms, uppercase letter may be used for readability (e.g., NORMACA). Normal case is used when the contents of a field are being discussed. Fields containing only one bit are usually referred to as the name bit instead of the name field.

Numbers that are not immediately followed by lower-case b or h are decimal values.

Numbers immediately followed by lower-case b (xxb) are binary values.

Numbers or upper case letters immediately followed by lower-case h (xxh) are hexadecimal values.

The most significant bit of a binary quantity is shown on the left side and represents the highest algebraic value position in the quantity.

If a field is specified as not meaningful or the field is to be ignored, the entity that receives the field shall not take any action based on the value of that field.

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

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

The ISO convention of numbering is used (i.e., the thousands and higher multiples are separated by a space and a comma is used as the decimal point as in 65 536 or 0,5).

3.5 Notation Conventions

3.5.1 Notation for state diagrams

All state diagrams use the notation shown in figure 2.
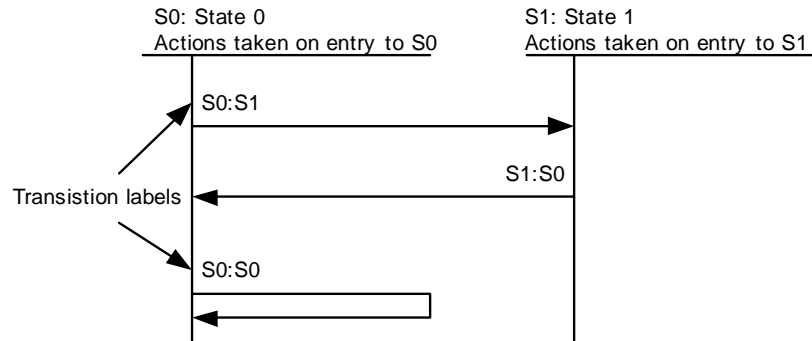
Figure 2 — Example state diagram

The state diagram is followed by a list of the state transitions, using the transition labels. Each transition is described in the list with particular attention to the conditions that cause the transition to occur and special conditions related to the transition. Using figure 2 as an example, the transition list might read as follows:

Transition S0:S1: This transition occurs when state S0 is exited and state S1 is entered.

Transition S1:S0: This transition occurs when state S1 is exited and state S0 is entered.

Transition S0:S0: This transition occurs when state S0 transitions to itself. It is particularly important to note that the actions taken whenever state S0 is entered are repeated every time this transition occurs.

A system specified in this manner has the following properties:

  a) time elapses only within discrete states;
  b) state transitions are logically instantaneous; and
  c) every time a state is entered, the actions of that state are started. Note that this means that a transition that points back to the same state restarts the actions from the beginning.

4 General Concepts

4.1 Overview

The SCSI stream device class specifies the behavior of a logical unit that is primarily a streaming data device. Two device types are members of this class: sequential-access and printer devices. This standard addresses the sequential-access device type only. The sequential-access device type has the characteristic of primarily handling data in a sequential manner (i.e., a stream). This does not limit the device's ability to position randomly within the data although a sequential-access device is not truly random-access (see SBC-2 for a description of a random-access device).

This standard describes two modes and associated command sets for communicating with a sequential-access device:

a) implicit address mode. Commands to read and write on a sequential-access device do not contain any positioning information fields. Instead, the device position is normally determined by previous commands; and

b) explicit address mode. Commands to read and write on a sequential-access device contain positioning information fields.

Commands are available for absolute and relative positioning. Writing to a sequential-access device may cause all data starting at the point at which the data is written to be invalidated. There may be restrictions on where write operations may be initiated. Reading or writing data as a long string of data, as in a stream, tends to be the most efficient.

4.2 Sequential-access device model

4.2.1 Sequential-access device model overview

Sequential-access devices are described herein from the point of view of a tape device. However, other implementations are not precluded.

Sequential-access devices optimize their use in storing or retrieving user data in a sequential manner. Since access is sequential, position changes typically take a long time, when compared to random-access devices.

4.2.2 Physical elements

The recording medium for tape devices consists of various widths and lengths of a flexible substrate coated with a semi-permanent magnetic material. The recording medium may be wound onto single reels or encapsulated into cartridges containing both a supply reel and a take-up reel. Several American National Standards exist covering the construction of reels and cartridges for interchange as well as recording techniques for many of the format or density combinations.

For a sequential-access device, a recording medium exists between two reels, the supply reel and take-up reel. The read/write mechanism may only access the medium between the reels. As the medium is taken out of one reel, it passes by the read/write mechanism and into the other reel. Transferring data as a stream is most efficient, since the medium may traverse the read/write mechanism producing a flow of data. To position to a given point requires moving the medium until the appropriate position is found.

The recording medium has two physical attributes called beginning-of-medium (BOM) and end-of-medium (EOM). Beginning-of-medium is at the end of the medium that is attached to the take-up reel. End-of-medium is at the end of the medium that is attached to the supply reel. In some cases, the medium is permanently affixed to one or both of the reel hubs. Beginning or end of medium is not required to be related to the beginning or end of any partition.

A volume is composed of the recording medium and its physical carrier (e.g., reel, cartridge, cassette). Volumes have an attribute of being mounted or de-mounted on a suitable transport mechanism.

Mounted is the state of a volume when the device is physically capable of processing commands that cause the medium to be moved. A volume is de-mounted when it is being loaded, threaded, unloaded, unthreaded, or when not attached to the device.

Ready is the state of the logical unit when medium access and non-medium access commands may be processed. The logical unit is not ready when no volume is mounted or, from the initiator's perspective, whenever any medium access command reports CHECK CONDITION status and a NOT READY sense key. The logical unit is not ready during the transition from mounted to not mounted, or not mounted to mounted. Devices may have a physical control that places the device in a not ready state even when a volume is mounted.

As shown in figure 3, a portion of the physical length of medium is not usable for recording data. For most volumes, a length of the medium is reserved between the take-up reel and the beginning-of-medium, and between the end-of-medium position and the supply reel. This is done to provide a sufficient tape wrap onto the reel hub and to ensure that recording starts in an undamaged section of the medium.
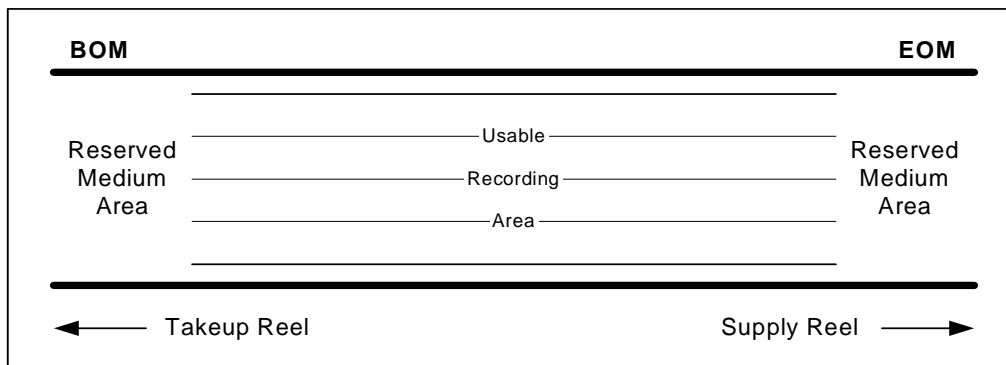
Figure 3 — Typical volume layout

The position on the medium where a pattern of recorded signals may be written by one write component is called a track (see figure 4). A device may write or read from one or more tracks at a time, depending on the format.
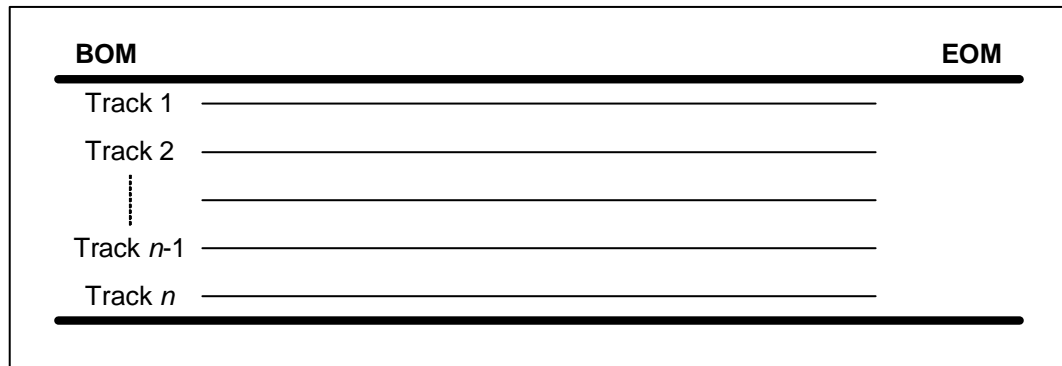
Figure 4 — Typical medium track layout

On a new volume, recording of one or more tracks begins after mounting the volume and moves from beginning-of-medium toward end-of-medium. The number of tracks written at one time is called a track group (TrkGrp). Track groups may be used by any recording format. For recorded volumes, reading in the forward direction follows the same course of tracks when writing.

In serpentine recording, not all tracks are recorded at the same time. At the end-of-medium or beginning-of-medium, the device reverses direction and begins recording the next track group. The process of reversing direction and recording the next track group may be repeated until all track groups are recorded. For serpentine devices that record only one track at a time, each physical track represents one track group (see figure 5).
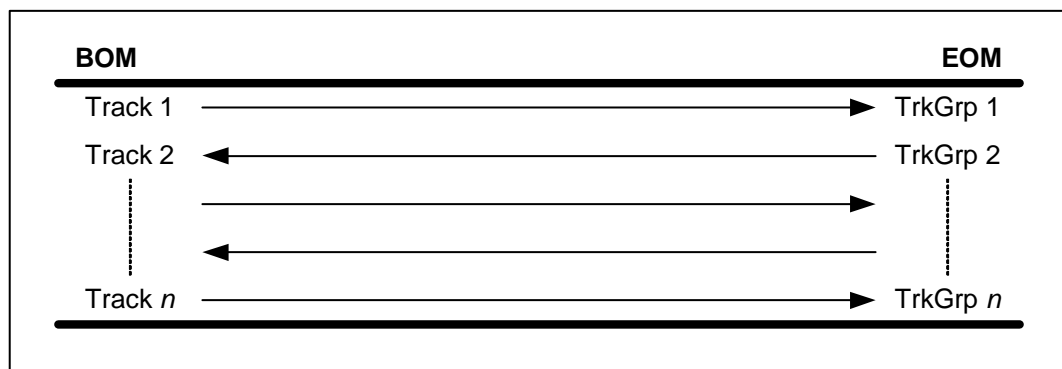
Figure 5 — Serpentine recording example

Some multi-track devices have only one track group, using a parallel storage format that supports the simultaneous recording of all available tracks (see figure 6).
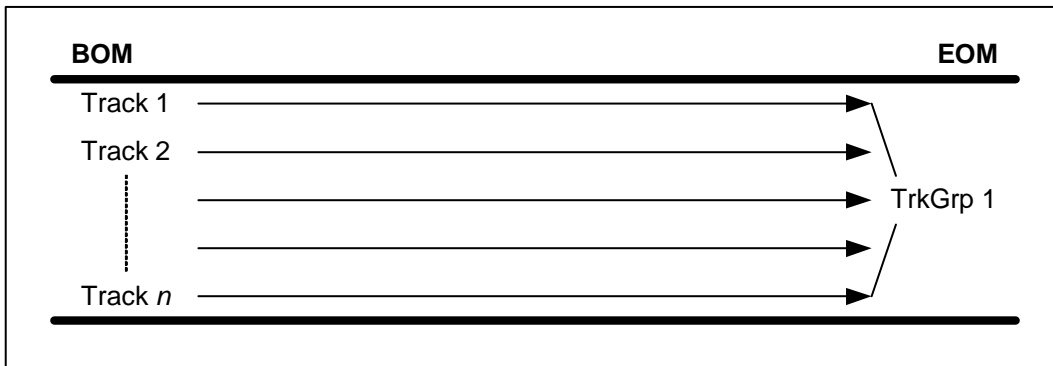
Figure 6 — Parallel recording example

The serpentine and parallel recording formats shown in the previous examples define tracks as longitudinal patterns of recorded information. One other storage format used by some devices records tracks diagonally across the medium. One or more tracks may be recorded at the same time. This recording technique is known as helical scan (see figure 7).
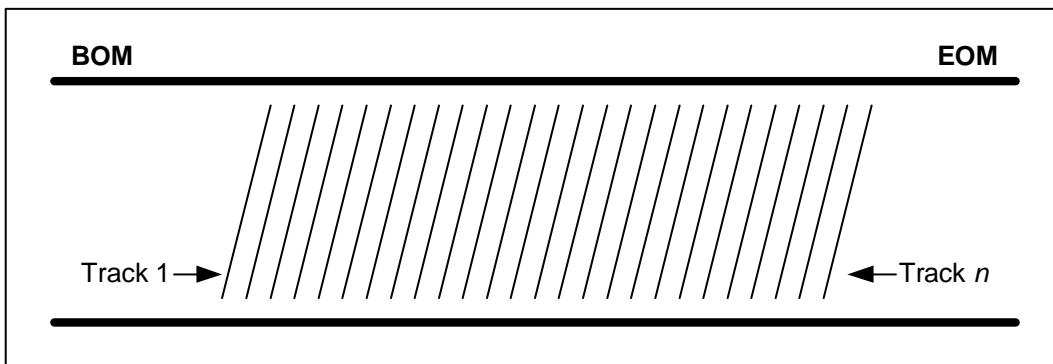
Figure 7 — Helical scan recording example

For most recording formats, a format identification in the form of a tone burst or some other recognizable pattern is recorded outside the user data area. The format identification is an attribute of a volume used for interchange purposes and is defined in applicable standards.

4.2.3 Early-warning

When writing, the application client needs an indication that it is approaching the end of the permissible recording area when moving in a direction toward the end of the partition (see 4.2.4). This position, called early-warning (EW), is typically reported to the application client at a position early enough for the device to write any buffered logical objects to the medium while still leaving enough room for additional recorded logical objects (see figure 8). Some American National Standards include physical requirements for a marker placed on the medium to be detected by the device as early-warning.
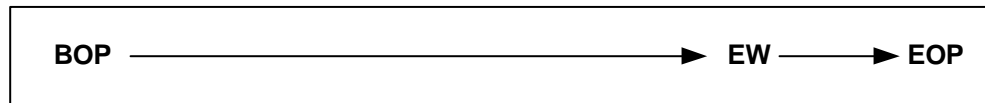
Figure 8 — Early-warning example

Devices are expected to report early warning to the application client when sufficient recording space is nominally available before EOP to record logical objects in the object buffer(s) and some additional logical objects. A logical concept of early-warning may be required to signal the application client at an appropriate location prior to the physical marker, particularly for devices that implement object buffers.

4.2.4 Partitions within a volume

Partitions consist of one or more non-overlapped logical volumes, each with its own beginning and ending points, contained within single physical volume. Each partition (*x*) within a volume has a defined beginning-of-partition (BOP *x*), an early-warning position (EW *x*), and an end-of-partition (EOP *x*).

All volumes have a minimum of one partition called partition 0, the default data partition. For devices that support only one partition, the beginning-of-partition zero (BOP 0) may be equivalent to the beginning-of-medium and the end-of-partition zero (EOP 0) may be equivalent to the end-of-medium. For devices that support more than one partition, they shall be numbered sequentially starting with zero (i.e., beginning-of-partition 0).

When a volume is mounted, it is logically positioned to the beginning of the default data partition (BOP 0). When a REWIND command is received in any partition (*x*), the device positions to the beginning-of-partition of the current partition (BOP *x*).

Partitions on a volume may be recorded in any order and use any partition number unique to the physical volume. It is sufficient for a device to be able to locate a partition (given its partition number) or to determine that it does or does not exist on the volume. For interchange, information about which partitions are present on a volume may be stored on the volume in a format specified area (possibly unavailable to the application client) or the information may be an intrinsic attribute of the device implementation.

Figure 9 shows a possible partition implementation for a four-track serpentine recording device, assuming that each track group defines a partition.
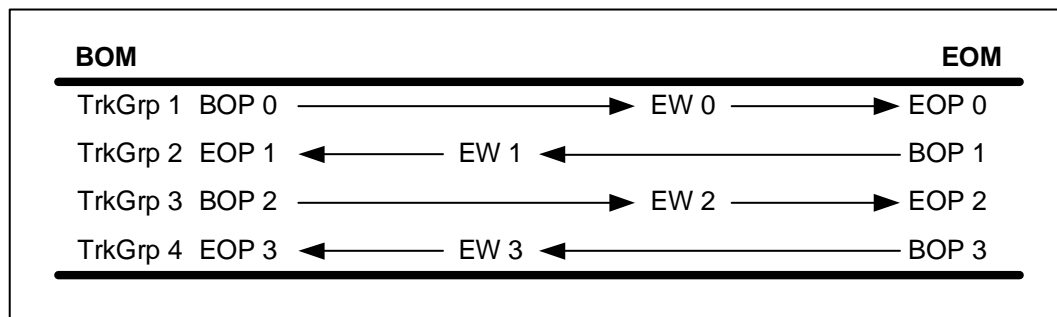
Figure 9 — Partitioning example - one partition per track group

Another possible partition implementation for this four-track serpentine recording device is shown in figure 10, using two track groups to define each partition.
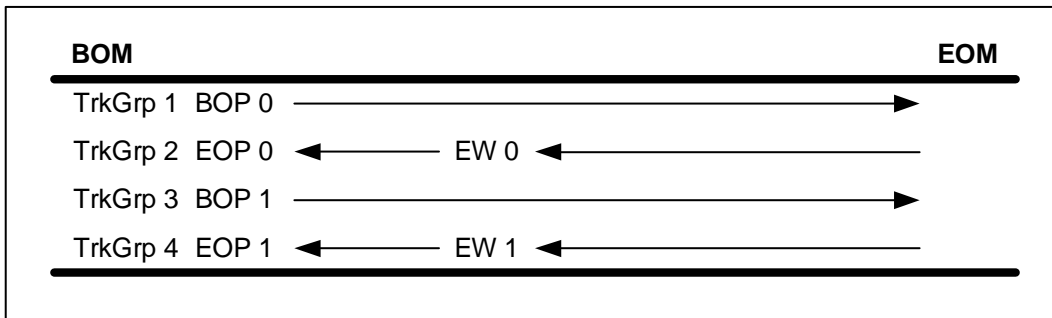
Figure 10 — Partitioning example - one partition per two track groups

The previous examples show the beginning and ending points for a partition aligned with physical bounds of the medium. This is not a mandatory requirement for partitioning. It is sufficient for a device to be able to locate to and stay in any partition bounded by a BOP *x* and EOP *x*. In this case, a device-recognizable attribute could be used to delineate the partitions. Figure 11 shows a possible two-partition implementation for a device with only one track group.
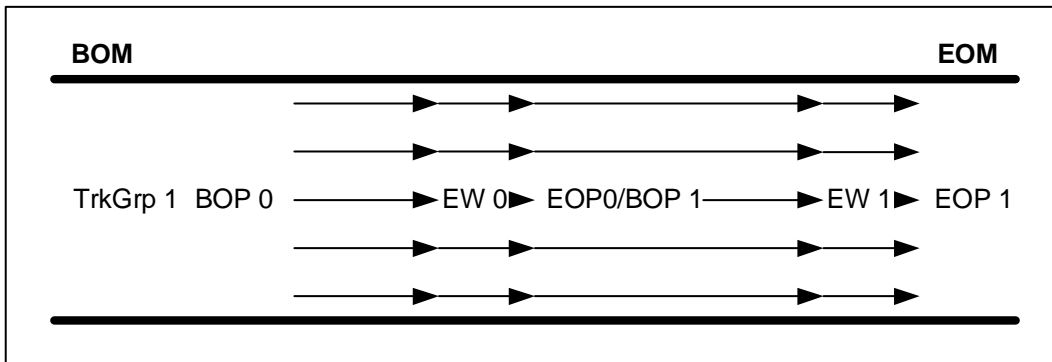
Figure 11 — Partitioning example - two partitions per track group

Three methods are defined in the MODE SENSE and MODE SELECT commands for managing partitions:

a) device-defined fixed locations;
b) device-defined based on an application client supplied number of partitions and a vendor-specific allocation algorithm; and
c) definition by partition number and capacity by an application client.

4.2.5 Logical objects

4.2.5.1 Logical objects within a partition

The area between BOP *x* and EOP *x* on a recorded volume contains application client accessible logical objects. Logical objects are controlled and transferred between the application client and the medium using commands defined in this standard. Each logical object shall have a logical object identifier that is unique within a partition.

The basic unit of data transferred by an application client is called a logical block. Logical blocks are stored according to the specifications of the format for the volume and may be recorded as portions of one or more physical blocks on the medium. The mapping between physical and logical blocks is the responsibility of the device server.

Filemarks are special recorded logical objects not containing user data. Proper recording and detection of filemarks is the responsibility of the device server. Application clients traditionally use filemarks to separate groups of user data from each other. Since some format standards do not define an explicit EOD, operating system software has often used conventions with filemarks to represent an EOD indication. In some implementations, the device's EOD definition may be specified by the application client using the Device Configuration mode page (see 8.3.3).

Inter-block gaps, the gaps between logical objects, are introduced on the medium at the time a logical object is written without explicit action by the application client. Minimum and maximum lengths for inter-block gaps are defined by the recording format.

Erase gaps may be recorded on the medium through use of the ERASE command or device-initiated error recovery actions. Although explicitly recorded on the medium, there is normally no distinction between two contiguous erase gaps. An erase gap may be a length of erased medium or a recorded pattern not distinguishable as a logical object. Minimum and maximum lengths for erase gaps are defined by the recording format.

After writing data from BOP *x*, the medium is considered to be a contiguous grouping of logical objects and gaps. Certain American National Standards define gap lengths that, if exceeded, are to be considered as having reached blank medium (i.e., end of data within a partition). Depending on the format, this blank medium may be treated as an end-of-data indication, an error recovery area, or an unrecoverable medium error causing an interchange error. Unrecorded volumes, new or erased, may exhibit blank medium characteristics if an attempt is made to read or space the volume before data has been written.

A sequential-access device may be capable of supporting fixed-block transfers or variable-block transfers. The concept of fixed or variable mode for writing and reading logical blocks only specifies the method by which the application client specifies the size of a logical block for transfer, and not the method of recording physical blocks on the medium. However, a device that supports only fixed-length physical blocks may only be capable of supporting logical blocks of the same length. The length of a logical block is always described in bytes. Refer to the READ BLOCK LIMITS command (see 7.4) for additional information about fixed-block transfers and variable-block transfers.

4.2.5.2 Logical object identifier

The logical object identifier value shall be a sequentially increasing number assigned to each logical object recorded in the partition starting with zero for the recorded logical object at BOP.

The READ POSITION command may be used to determine a logical object identifier and the application client may use this value with a LOCATE command or an explicit command to position to the same location at some future time.

4.2.6 Logical files

4.2.6.1 Logical files within a partition

Application clients may use filemarks to separate groups of user data into logical files. A logical file shall contain zero or more logical blocks and shall begin with the next logical object following a filemark or BOP. Each logical file shall have a logical file identifier that is unique within the partition.

4.2.6.2 Logical file identifier

The logical file identifier value shall be a sequentially increasing number assigned to each logical file recorded in the partition starting with zero for the recorded logical file beginning at BOP.

The READ POSITION command may be used to determine a logical file identifier and the application client may use this value with a LOCATE(16) command to position to the BOP side of the same logical file at some future time.

4.2.7 Object buffering

A device may contain a temporary storage area capable of holding one or more logical objects - an object buffer. A device object buffer may include any combination of logical objects in the process of being written to the medium, or it may contain read-ahead logical objects transferred from the medium.

A device with an object buffer may be capable of operating in either a buffered mode or an unbuffered mode. A device with no object buffer operates only in unbuffered mode. Either term is only applicable to the manner in which the device manages information to be written to the medium. Buffered mode is not applicable during read commands, regardless of whether read data passes through an object buffer.

A device operating in buffered mode may return GOOD status for write operations when all logical objects have been successfully transferred from the application client into the device object buffer. For devices operating in unbuffered mode, GOOD status is not returned until all requested logical objects are successfully recorded on the medium.

When issuing a buffered WRITE FILEMARKS command with the immediate bit set to one, GOOD status shall be returned as soon as the command is validated. For a WRITE FILEMARKS command with the immediate bit set to zero, the device server shall perform a synchronize operation (see 4.2.8).

If an unrecoverable write error occurs while in buffered mode, the device generates an error condition to the current active command. If no command is active, the error may be reported on the next applicable operation as a deferred error (see SPC-3). For some implementations auto contingent allegiance may be required. Refer to SAM-3 for a description auto contingent allegiance.

The READ POSITION command may be used to determine the number and storage space of buffered logical objects not written before the unrecoverable error was encountered.

A device that encounters an unrecoverable error during a read-ahead operation shall not report the error unless the logical object in error is accessed by an application client.

Prior to performing some commands, the device server shall perform a synchronize operation (see 4.2.8) as stated in table 12 and table 19.

4.2.8 Synchronize operation behavior

As stated in table 12 and table 19, some commands may require the device server to perform a synchronize operation (see 3.1.64). If a command requires a synchronize operation, the synchronize operation shall be performed prior to initiating any command-specific operations. Upon successful completion of the synchronize operation, no logical objects shall remain in the object buffer that have not been written to the medium. A synchronize operation shall have no effect on an object buffer that contains only read-ahead logical objects, or logical objects that have already been successfully written to the medium.

For a WRITE BUFFER command specifying modes 4, 5, 6, or 7 (download microcode operations), the device server shall perform a synchronize operation before performing the download operation.

For a MODE SELECT command specifying the Medium Partition mode page, the device server shall perform a synchronize operation before the logical unit partitions the medium.

For a SEND DIAGNOSTICS command, the device server shall perform a synchronization operation before any diagnostic tests that may affect the buffered logical objects, media, or logical position, are initiated.

4.2.9 Direction and position definitions

For sequential-access devices, positioning has the connotation of logically being in, at, before, or after some defined place within a volume. Positioning requires that the position is capable of being repeated under the same circumstances. The orientation of usage for the four words (in, at, before, or after) is in one direction, from BOP *x* toward EOP *x*. All positioning defined below is worded from this perspective. Devices without object buffers have some physical position that relates to these logical positions. However, these definitions do not require the medium to have a physical position equivalent to the logical position unless explicitly stated.

The forward direction is defined as logically progressing from BOP *x* toward EOP *x*. The reverse direction is defined as logically progressing from EOP *x* toward BOP *x*. In serpentine devices, the logical forward or reverse direction has an alternating relationship to the physical motion of the medium.

The concept of being in some position means not being outside a defined region. The definition allows the position to be on the boundary of a defined region. When a volume is first mounted, the logical position is always at the beginning of the default data partition (BOP 0). Whenever a volume is mounted and the medium motion is stopped, the position is in some partition. While moving between partitions, there is no stable position.

The concept of being at some position specifies being positioned to a logical or physical extremity of a partition. A sequential-access device may be positioned at BOM, at BOP *x*, at EOD, at EOP *x*, or EOM, since these are stable positions at extremities of a partition.

The concept of being before some position specifies that there is some logical object or other defined point that may be encountered when moving toward EOP *x*, if the proper commands are issued. Being positioned before a particular logical block means that if the device receives a valid READ command, the logical block is transferred to the application client. This position may also be before EW *x* and EOP *x*, since these are defined points within any partition. However, if data has not been written to the end-of-partition, these points may not be accessible by the initiator.

The concept of being after some position specifies that there is some logical object or other defined point on the BOP *x* side of the current position that may be encountered if the proper commands are issued. When a READ command for a single logical block has been successfully processed, the logical position is after the transferred logical block.

4.2.10 Error reporting

4.2.10.1 Overview

Sequential-access devices compliant with this standard shall support both the fixed and descriptor sense data formats (see SPC-3). If fixed format sense data is specified, but the INFORMATION field value exceeds the maximum value allowed in the fixed format sense data, the VALID bit shall be set to zero. Refer to SPC-3 for a description of the sense data VALID bit and INFORMATION field contained in the REQUEST SENSE sense data. In addition, this standard describes the use of the INFORMATION field specific to the sequential-access device type.

4.2.10.2 Stream commands sense data descriptor

Table 1 defines the streams commands sense data descriptor for sequential-access devices.

Table 1 — Stream commands sense data descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (04h)							
1	ADDITIONAL LENGTH (02h)							
2	Reserved							
3	FILEMARK	EOM	ILI	Reserved				

See SPC-3 for a description of the DESCRIPTOR TYPE and ADDITIONAL LENGTH fields.

See the READ(16) (see 5.3), READ(6) (see 6.4), SPACE(6) (see 6.6), RECOVER BUFFERED DATA (see 7.6), and SPACE(16) (see 7.10) commands for a description of FILEMARK bit usage.

See the READ(16) (see 5.3), READ REVERSE(16) (see 5.4), WRITE(16) (see 5.6), WRITE FILEMARKS(16) (see 5.7), LOCATE(10) (see 6.3), READ(6) (see 6.4), READ REVERSE(6) (see 6.5), SPACE(6) (see 6.6), WRITE(6) (see 6.8), WRITE FILEMARKS(6) (see 6.9), LOCATE(16) (see 7.3), RECOVER BUFFERED DATA (see 7.6), and SPACE(16) (see 7.10) commands, and the Device Configuration mode page (see 8.3.3) for end-of-medium (EOM) bit usage.

See the READ(16) (see 5.3) and READ(6) (see 6.4) commands and the Data Compression mode page (see 8.3.2) for incorrect length indicator (ILI) bit usage.

4.2.10.3 Error conditions

If any of the following conditions occur during the processing of a command or if a deferred error prevented the command from processing, the device server shall return CHECK CONDITION status. The appropriate sense key and additional sense code should be set. Table 2 illustrates some error conditions and the applicable sense keys. Table 2 does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Table 2 — Error conditions and sense keys

Condition	Sense Key
Unsupported option requested.	ILLEGAL REQUEST
Logical unit reset, I_T nexus loss, or medium change since last command from this initiator.	UNIT ATTENTION

Table 2 — Error conditions and sense keys (Continued)

Condition	Sense Key
Self diagnostic failed.	HARDWARE ERROR
Unrecovered read error.	MEDIUM ERROR HARDWARE ERROR
Recovered read or write error.	RECOVERED ERROR
Attempt a WRITE, READ, READ REVERSE, VERIFY, or RECOVER BUFFERED DATA command with the FIXED bit set to zero and variable-block transfers are not supported.	ILLEGAL REQUEST
Attempt a WRITE, READ, READ REVERSE, VERIFY, or RECOVER BUFFERED DATA command with the FIXED bit set to zero and requested block length is not supported.	ILLEGAL REQUEST
Attempt a WRITE, READ, READ REVERSE, VERIFY, or RECOVER BUFFERED DATA command with the FIXED bit set to one and MODE SENSE block length set to zero.	ILLEGAL REQUEST
Attempt to perform an erase, format, partition, set capacity, or write-type operation on write protected medium.	DATA PROTECT
Deferred write error.	MEDIUM ERROR VOLUME OVERFLOW HARDWARE ERROR

See the READ(16) (see 5.3), READ REVERSE(16) (see 5.4), VERIFY(16) (see 5.5), WRITE(16) (see 5.6), READ(6) (see 6.4), READ REVERSE(6) (see 6.5), VERIFY(6) (see 6.7), WRITE(6) (see 6.8), and RECOVER BUFFERED DATA (see 7.6) commands for a description of the FIXED bit.

The Read-Write Error Recovery mode page (see 8.3.5) current values specify behavior when an unrecoverable read or write error is encountered. If the Read-Write Error Recovery mode page is not implemented, the behavior is vendor-specific.

In the case of a deferred write error, the sense data VALID bit shall be set to zero.

In the case of an unrecovered write error or a deferred write error, if buffered mode 1h is selected, the error shall be reported to the first application client issuing a command (other than INQUIRY or REQUEST SENSE). If buffered mode 2h is selected, the error shall be reported to the initiator with unwritten data in the object buffer.

In the case of a write attempt to write protected medium, the additional sense code specifies the cause of the DATA PROTECT sense key (see 4.2.11).

4.2.11 Write protection

4.2.11.1 Write protection introduction

Write protection of the medium prevents the alteration of logical objects on the medium and any change to the accessibility of logical objects on the medium, by commands issued to the device server. Write protection is usually controlled by the user of the medium through manual intervention (e.g., mechanical lock) or may result from hardware controls (such as tabs on the media housing), conditions such as positioning within unrecoverable data, or software write protections. All sources of write protection are independent. When present, any write protection shall cause otherwise valid commands that request alteration of logical objects on the medium, or affect the accessibility of logical objects on the medium, to be rejected with a CHECK CONDITION status with the sense key set to

DATA PROTECT (see 4.2.11.2). Only when all write protections are disabled shall the device server process commands that request alteration of logical objects on the medium, or commands that may affect the accessibility of logical objects on the medium.

Hardware write protection results when a physical attribute of the drive or medium is changed to specify that writing shall be prohibited. Changing the state of the hardware write protection requires physical intervention, either with the drive or the medium. If allowed by the drive, changing the hardware write protection while the medium is mounted results in vendor-specific behavior that may include the writing of previously buffered logical objects.

Conditions such as positioning within unrecoverable data may result in a temporary write protection condition. To preserve future data integrity, the device server may reject any command that requires writing data to the medium when the recovery of the data is uncertain. A temporary write protection condition may be released by the device server at any time. Buffered logical objects may or may not be written to the media (e.g., the initiator unloads the volume before the temporary write protection condition is removed). The exact behavior of the device server during a temporary write protection condition is vendor-specific.

Software write protection results when either the device server or medium is marked as write protected by a command from the application client. Four optional means of setting a software write protection state are available to an application client through the Device Configuration and Control mode pages:

   a) software write protection for the device server across mounts;
   b) associated write protection for the currently mounted volume;
   c) persistent write protection of a volume across mounts; and
   d) permanent write protection of a volume across mounts.

The application client may control these write protections using the MODE SELECT command with the Control mode page (see SPC-3) and the Device Configuration mode page (see 8.3.3). All of the software write protection methods are optional. Changing the state of any software write protection shall not prevent previously buffered logical objects from transferring to the media.

4.2.11.2 Write protection additional sense code use

The additional sense code associated with the DATA PROTECT sense key depends on the write protection in effect at the time. Table 3 specifies the preferred additional sense code for the given write protection. Alternatively, the generic additional sense code of WRITE PROTECTED may be returned by the device server.

Table 3 — Write protect additional sense code combinations

Cause of DATA PROTECT error	Additional Sense Code
Hardware Write Protection	HARDWARE WRITE PROTECTED
Permanent Write Protection	PERMANENT WRITE PROTECT
Persistent Write Protection	PERSISTENT WRITE PROTECT
Associated Write Protection	ASSOCIATED WRITE PROTECT
Software Write Protection	LOGICAL UNIT SOFTWARE WRITE PROTECTED

If more than one condition exists, the device server shall either report the applicable condition in order of HARDWARE WRITE PROTECTED, PERMANENT WRITE PROTECT, PERSISTENT WRITE PROTECT, ASSOCIATED WRITE PROTECT, and LOGICAL UNIT SOFTWARE WRITE PROTECTED, or report the generic additional sense code of WRITE PROTECTED.

Other conditions that can cause a command to be rejected with a DATA PROTECT sense key include:

a)   the format on the current medium is read-only by the device server;
b)   the device server can only write from BOP or EOD and the current logical position is neither;
c)   the medium is an archive tape and only can be recorded at EOD; and
d)   vendor-specific conditions.

4.2.11.3 Software write protection for the device server

Software write protection for the device server controls write protection for the device server. This method of write protection is optionally controlled from the Control mode page (see SPC-3) or the SWP bit in the Device Configuration mode page (see 8.3.3). Either or both methods may be implemented by the device server. If both methods are implemented, each control bit is independently set. Software write protection exists if either bit is non-zero. The state of software write protection for the device server shall not be recorded on media. The value of the SWP bit may be altered by the application client (if the SWP bit is changeable). The state of each control bit shall be set to its default state after a logical unit reset.

4.2.11.4 Associated write protection

Associated write protection controls write protection for the currently mounted volume as long as the current volume is mounted. The associated write protection state is controlled by the ASOCWP bit in the Device Configuration mode page (see 8.3.3). Associated write protection exists if the ASOCWP bit is non-zero. Associated write protection may be altered by the application client (if the ASOCWP bit is changeable) if a volume is mounted. If a volume is de-mounted or after a logical unit reset occurs, associated write protection shall be removed.

4.2.11.5 Persistent write protection

Persistent write protection controls write protection for the currently mounted volume. The persistent write protection state is controlled by the PERSWP bit in the Device Configuration mode page (see 8.3.3). If enabled, persistent write protection shall exist for the mounted volume until disabled by the application client. The state of persistent write protection shall be recorded with the volume and the persistent write protection shall only affect the application client accessible medium. The device server shall report the PERSWP bit as one when a mounted volume is marked with persistent write protection. If a volume is de-mounted or after a logical unit reset occurs, the device server shall report the PERSWP bit as zero prior to the mounting of a volume. The means for recording the state of persistent write protection for the volume may be specified in the applicable recording format standard or be vendor-specific.

4.2.11.6 Permanent write protection

Permanent write protection controls write protection for the currently mounted volume. The permanent write protection state is controlled by the PRMWP bit in the Device Configuration mode page (see 8.3.3). If enabled, permanent write protection shall exist for the mounted volume until disabled by a vendor-specific method. The state of permanent write protection shall be recorded with the volume and the persistent write protection shall only affect the application client accessible medium. The device server shall report the PRMWP bit as one when a mounted volume is marked with permanent write protection. If a volume is de-mounted or after a logical unit reset occurs, the device server shall report the PRMWP bit as zero prior to the mounting of a volume. The means for recording the state of permanent write protection for the volume may be specified in the applicable recording format standard or be vendor-specific. Permanent write protection shall not be removed by a MODE SELECT command using the PRMWP bit. Methods to remove this protection may or may not exist and are vendor-specific.

4.2.12 Progress indication

For the following immediate operations where the device server remains ready, an application client may test the progress of the operation (see table 4).

Table 4 — Commands providing progress indication without changing ready state

Operation	Options	Subclause	Additional Sense Code
ERASE	LONG = 1	5.2,6.2	ERASE OPERATION IN PROGRESS
LOCATE		5.2,6.3	LOCATE OPERATION IN PROGRESS
REWIND		7.8	REWIND OPERATION IN PROGRESS
SET CAPACITY		7.9	SET CAPACITY OPERATION IN PROGRESS
VERIFY		5.5,6.7	VERIFY OPERATION IN PROGRESS

While the device server is performing the operation, an application client may test the progress of the operation by interpreting the progress indication information in the sense-key specific field of the sense data. During the operation, the device server may report a sense key value of NO SENSE and additional sense code as specified in table 4. The device server should use the sense key specific function for progress indication to provide information on the completion of the operation.

For the following immediate operations where the device server is ready or will become ready, an application client may follow the progress of the operation (see table 5).

Table 5 — Commands changing ready state and providing progress indication

Operation	Options	Subclause	Additional Sense Code
FORMAT MEDIUM		7.1	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS
LOAD UNLOAD	LOAD = 1, EOT = 0	7.2	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
LOAD UNLOAD	LOAD = 0, EOT = 1	7.2	LOGICAL UNIT NOT READY, OPERATION IN PROGRESS

While the device server is performing the operation, an application client may test the progress of the operation by interpreting the progress indication information in the sense-key specific field of the sense data. During the operation, the device server may report a sense key value of NOT READY and an additional sense code as specified in table 5. The sense key specific function for progress indication may be used by the device server to provide information on the completion of the operation.

> NOTE 1 A REQUEST SENSE command following a TEST UNIT READY command that results in CHECK CONDITION status may provide information that, if acted upon, may lead to unexpected conditions. For example, progress indication reporting is useful when a media changer is used to service a sequential-access device following an unload operation with IMMED=1b. A TEST UNIT READY command may receive CHECK CONDITION status and a NOT READY sense key reported in the subsequent sense data. This may imply that the unload operation is finished. If the initiator ignores the progress indication information in the sense data, an EXCHANGE MEDIUM or MOVE MEDIUM command (see SMC-2) to move the dismounted volume from the device may fail to grab the volume if the unload operation is still in progress.

4.2.13 Tagged command queuing

4.2.13.1 Tagged command queuing overview

A device server may optionally choose to implement support for tagged tasks (i.e., command queuing). Issuing tagged write commands with object buffering disabled can facilitate streaming operations up to the limit of the number of outstanding tagged commands supported by the application client and the device server. This limit may effectively reduce the usable portion of the object buffer which may affect device server performance.

> NOTE 2 For proper operation when performing tagged command queuing operations, an application client should wait for status to be returned for any MODE SELECT command before issuing the next command.

> NOTE 3 Since the EXTENDED COPY command does not specify an immediate bit, the use of tagged command queuing may be required to prevent an overlapped command condition (e.g., when querying the progress of the extended copy operation using the RECEIVE COPY RESULTS command).

4.2.13.2 Explicit address mode tagged write sequences

When operating in explicit address mode, tagged write sequences (see 3.1.65) are used to support tagged command queuing for write operations.

For explicit address mode tagged write sequences, the following rules apply:

  a) for a tagged write sequence consisting of more than one command, the FCS bit (see 5.2, 5.6, 5.7) shall be set to one in the first command of the tagged write sequence. For all other commands within the tagged write sequence, the FCS bit shall be set to zero;
  b) for a tagged write sequence consisting of more than one command, the LCS bit (see 5.2, 5.6, 5.7) shall be set to one in the last command of the tagged write sequence. For all other commands within the tagged write sequence, the LCS bit shall be set to zero;
  c) for a tagged write sequence consisting of only one command, the FCS bit and LCS bit shall be set to one;
  d) for a tagged write sequence consisting of more than one command, the application client shall issue the commands in sequentially-increasing logical object identifier order;
  e) an application client shall not issue a tagged write sequence prior to receiving status for all outstanding read type commands; and
  f) an application client shall specify a Command Reference Number (see SAM-3) for each command in a tagged write sequence.

4.2.14 Block address mode

4.2.14.1 Block address mode overview

When operating in implicit address mode, spacing operations and commands to read and write on the medium do not contain positioning information (i.e., a command is processed based on the medium position relative to the last command that was processed). As such, the application client does not contain explicit knowledge of the medium position. If an error occurs, to maintain data integrity, the application client must determine the medium position before re-issuing a command that affects the medium position.

When operating in explicit address mode, commands to read and write on the medium contain positioning information fields (i.e., a command is processed based on the medium position information specified in the command). As such, the application client contains explicit knowledge of the medium position. This results in enhanced error detection and recovery functionality that allows the application client to maintain data integrity while performing various operations without first determining the medium position. Some example operations include:

  a) the re-issuing of a command that affects the medium position;

b)   multi-path I/O; and

c)   tagged command queuing.

The device server shall support explicit address mode only, implicit address mode only, or both explicit address mode and implicit address mode. At any instance, the device server shall be in or transitioning between one of the following block address mode states (see 4.2.14.3):

a)   A0:Idle;

b)   E0:Explicit Address Mode - Neutral;

c)   E1:Explicit Address Mode - Write Capable; or

d)   F0:Implicit Address Mode.

4.2.14.2 Block address mode selection

The block address mode shall be selected as follows:

a)   if the BAML bit in the Device Configuration mode page (see 8.3.3) is set to zero, then the setting of the BAM bit in the Device Configuration mode page (see 8.3.3) shall be ignored and the block address mode shall be determined based on the first block address mode unique command that is received after a successful load operation or successful completion of a command that positions the medium to BOP;

b)   if the BAML bit in the Device Configuration mode page is set to one and the BAM bit in the Device Configuration mode page is set to zero, the logical unit shall support implicit address mode; or

c)   if the BAML bit in the Device Configuration mode page is set to one and the BAM bit in the Device Configuration mode page is set to one, the logical unit shall support explicit address mode.

Prior to performing a block address mode change, the logical unit shall perform a synchronize operation (see 4.2.8).

If the device server receives a command that is not supported by the currently selected mode, the device server shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to:

a)   ILLEGAL COMMAND WHILE IN EXPLICIT ADDRESS MODE if the currently selected mode is the explicit address mode;

b)   ILLEGAL COMMAND WHILE IN IMPLICIT ADDRESS MODE if the currently selected mode is the implicit address mode; or

c)   ILLEGAL COMMAND WHILE IN WRITE CAPABLE STATE if the device server is in explicit address mode write capable state.

4.2.14.3 Block address mode state diagrams

For the block address mode state diagrams (see figure 13, figure 14, figure 15, and figure 16), the following terminology shall apply:

a)   explicit command: a command contained only in the explicit address command set (see table 12);

b)   implicit command: a command contained only in the implicit address command set (see table 19); and

c)   generic command: an explicit command that is not a read type or write type command (see table 12).

A common command containing a BAM bit (e.g., LOCATE(16)) shall be processed as either an explicit or implicit command based on the setting of the BAM bit contained in the common command.

The SPACE(16) command shall be processed as either an explicit or implicit command based on the setting of the PARAMETER LENGTH field in the command descriptor block.

Figure 12 provides an overview of the block address model state diagram. Refer to figure 13, figure 14, figure 15, and figure 16 for detailed descriptions of the block address model state diagram.
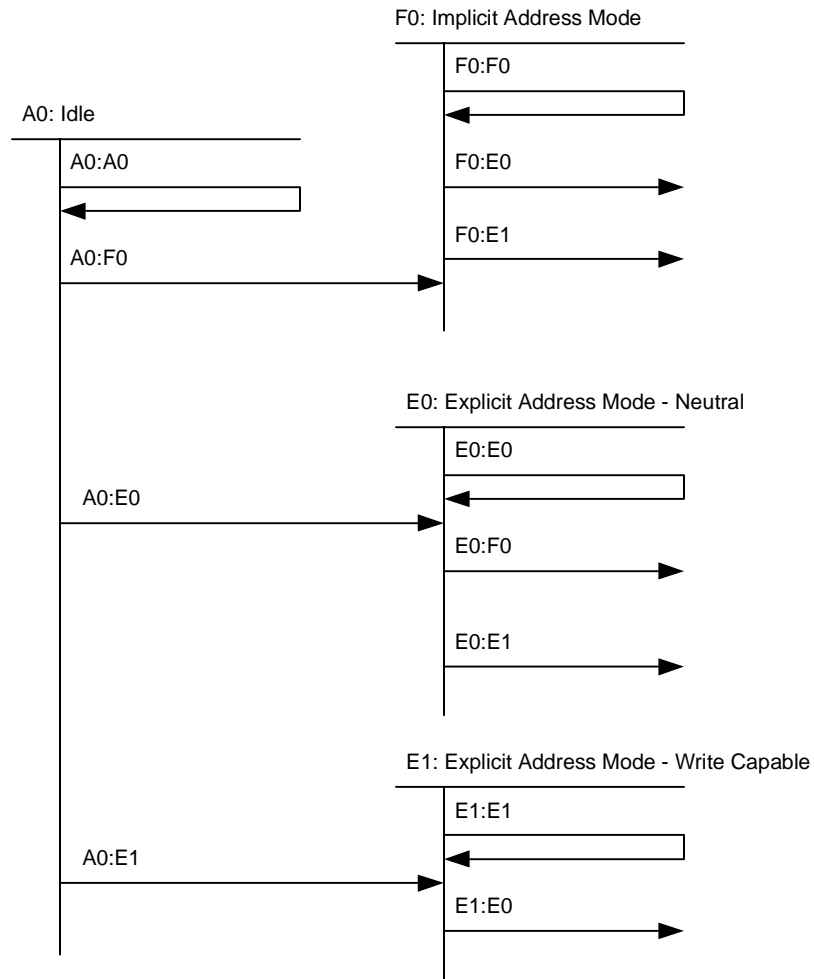
Figure 12 — Block address mode state diagram, overview

A0: Idle
Process enabled command

All:A0 Power on, BAML=0, BAM=0

A0:E0 MODE SELECT, BAML=1, BAM=1

A0:E0 Generic command enabled

All:A0 Logical unit reset, BAML=0, BAM=0

A0:E0 BOx=True, BAML=0, Explicit untagged command enabled

All:A0 I_T nexus loss, BAML=0, BAM=0

A0:E0 BOx=True, BAML=0, Explicit tagged read type command enabled

A0:E1 BOx=True, BAML=0, Explicit tagged write sequence command enabled, FCS=1

A0:F0 MODE SELECT, BAML=1, BAM=0

A0:F0 BAML=0, Implicit command enabled

A0:A0 FORMAT MEDIUM command enabled

A0:A0 MODE SELECT, non Device Configuration mode page

A0:A0 MODE SELECT, BAML=0, BAM=0

A0:A0 SEND DIAGNOSTIC command enabled

A0:A0 SET CAPACITY command enabled

A0:A0 Error, BOx=False, Explicit command enabled

A0:A0 Error, Explicit tagged write type command enabled, FCS=0

Figure 13 — Block address mode state diagram, Idle state

State A0:Idle: This is the idle state.

Transition All:A0: This transition shall occur when a power-on, logical unit reset, or I_T nexus loss event occurs and the BAML bit is set to zero and the BAM bit is set to zero.

Transition A0:E0: This transition shall occur when:

   a) a MODE SELECT command specifying a Device Configuration mode page with the BAML bit set to one and the BAM bit set to one completes with GOOD status;
   b) a generic command is enabled;
   c) an explicit untagged command is enabled, the medium position is at BOx, and the BAML bit is set to zero; or
   d) an explicit tagged read type command is enabled, the medium position is at BOx, and the BAML bit is set to zero.

Transition A0:E1: This transition shall occur when an explicit tagged write sequence command is enabled with the FCS bit set to one, the medium position is at BOx, and the BAML bit is set to zero.

Transition A0:F0: This transition shall occur when:

   a) a MODE SELECT command specifying a Device Configuration mode page with the BAML bit set to one and the BAM bit set to zero completes with GOOD status; or
   b) an implicit command is enabled, the medium position is at BOx, and the BAML bit is set to zero.

Transition A0:A0: This transition shall occur when:

   a) a FORMAT MEDIUM command is enabled;
   b) a MODE SELECT command specifying a mode page other than the Device Configuration mode page is enabled;
   c) a MODE SELECT command specifying a Device Configuration mode page with the BAML bit set to zero and the BAM bit set to zero is enabled;
   d) a SEND DIAGNOSTIC command is enabled;
   e) a SET CAPACITY command is enabled;
   f) an explicit command is enabled and the medium position is not at BOx. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to SEQUENTIAL POSITIONING ERROR; or
   g) an explicit tagged write type command is enabled with the FCS bit set to zero. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

E0: Explicit Address Mode - Neutral
Process enabled command

E0:F0 MODE SELECT, BAML=1, BAM=0

All:E0 Power on, BAML=1, BAM=1

E0:F0 BOx=True, BAML=0, Implicit command enabled

All:E0 Logical unit reset, BAML=1, BAM=1

E0:E1 Explicit tagged write sequence command enabled, FCS=1

All:E0 I_T nexus loss, BAML=1, BAM=1

E0:E0 Explicit tagged read type command enabled

E0:E0 Explicit untagged command enabled

E0:E0 Generic command enabled

E0:E0 FORMAT MEDIUM command enabled

E0:E0 MODE SELECT, non Device Configuration mode page

E0:E0 MODE SELECT, BAML=0, BAM=0

E0:E0 SEND DIAGNOSTIC command enabled

E0:E0 SET CAPACITY command enabled

E0:E0 Error, Explicit tagged write type command enabled, FCS=0

E0:E0 Error, BOx=False, BAML=0, Implicit command enabled

Figure 14 — Block address mode state diagram, Explicit Address Mode - Neutral

State E0:Explicit Address Mode - Neutral: This is the neutral state for explicit address mode.

Transition All:E0: This transition shall occur when a power-on, logical unit reset, or I_T nexus loss event occurs and the BAML bit is set to one and the BAM bit is set to one.

Transition E0:F0: This transition shall occur when:

   a) a MODE SELECT command specifying a Device Configuration mode page with the BAML bit set to one and the BAM bit set to zero completes with GOOD status; or
   b) an implicit command is enabled, the medium position is at BOx, and the BAML bit is set to zero.

Transition E0:E1: This transition shall occur when an explicit tagged write sequence command is enabled with the FCS bit set to one.

Transition E0:E0: This transition shall occur when:

   a) an explicit tagged read type command is enabled;
   b) an explicit untagged command is enabled;
   c) a generic command is enabled;
   d) a FORMAT MEDIUM command is enabled;
   e) a MODE SELECT command specifying a mode page other than the Device Configuration mode page is enabled;
   f) a MODE SELECT command specifying a Device Configuration mode page with the BAML bit set to zero and the BAM bit set to zero is enabled;
   g) a SEND DIAGNOSTIC command is enabled;
   h) a SET CAPACITY command is enabled;
   i) an explicit tagged write type command is enabled with the FCS bit set to zero. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB; or
   j) an implicit command is enabled, the medium position is not at BOx, and the BAML bit is set to zero. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN EXPLICIT ADDRESS MODE.

E1: Explicit Address Mode - Write Capable
Process enabled command

E1:E1 Explicit tagged write sequence command
enabled, FCS=0, LCS=0

E1:E1 SEND DIAGNOSTIC command enabled

E1:E1 Allowed command enabled

E1:E0 Explicit tagged write sequence command
complete, LCS=1

E1:E0 Error, Explicit tagged write sequence
command enabled, FCS=1

E1:E0 Error, Explicit untagged write sequence
command enabled

E1:E0 Error, FORMAT MEDIUM, MODE
SELECT, or SET CAPACITY command enabled

E1:E0 Error, Explicit read type command
enabled

E1:E0 Error, Implicit command enabled

E1:E0 Error, Non-allowed Generic command
enabled

E1:E0 Error, Explicit tagged write sequence
command with out-of-order logical object
identifer enabled

Figure 15 — Block address mode state diagram, Explicit Address Mode - Write Capable

State E1:Explicit Address Mode - Write Capable: This is the write capable state for explicit address mode.

Transition E1:E1: This transition shall occur when:

a) an explicit tagged write sequence command is enabled with the FCS bit set to zero and the LCS bit set to zero;
b) a SEND DIAGNOSTIC command is enabled; or
c) an allowed command (see table 12) is enabled.

Transition E1:E0: This transition shall occur when:

a) an explicit tagged write sequence command with the LCS bit set to one completed with GOOD status;
b) an explicit tagged write sequence command with the FCS bit set to one and the LCS bit set to one completed with GOOD status;
c) an explicit tagged write sequence command is enabled with the FCS bit set to one. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB;

d)   an explicit untagged write sequence command is enabled. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN WRITE CAPABLE STATE;

e)   a FORMAT MEDIUM, MODE SELECT, or SET CAPACITY command is enabled. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN WRITE CAPABLE STATE;

f)   an explicit read type command is enabled. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN WRITE CAPABLE STATE;

g)   an implicit command is enabled. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN WRITE CAPABLE STATE;

h)   a non-allowed generic command (see table 12) is enabled. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN WRITE CAPABLE STATE; or

i)   an explicit tagged write sequence command with an out-of-order logical object identifier is enabled. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

Following an error condition that does not result in a transition out of write capable state, the following commands may be issued to transition from write capable state to neutral state:

a)   a WRITE(16) command with the LCS bit set to one and the TRANSFER LENGTH field set to zero; or

b)   a WRITE FILEMARKS(16) command with the LCS bit set to one, the IMMED bit set to zero, and the TRANSFER LENGTH field set to zero.
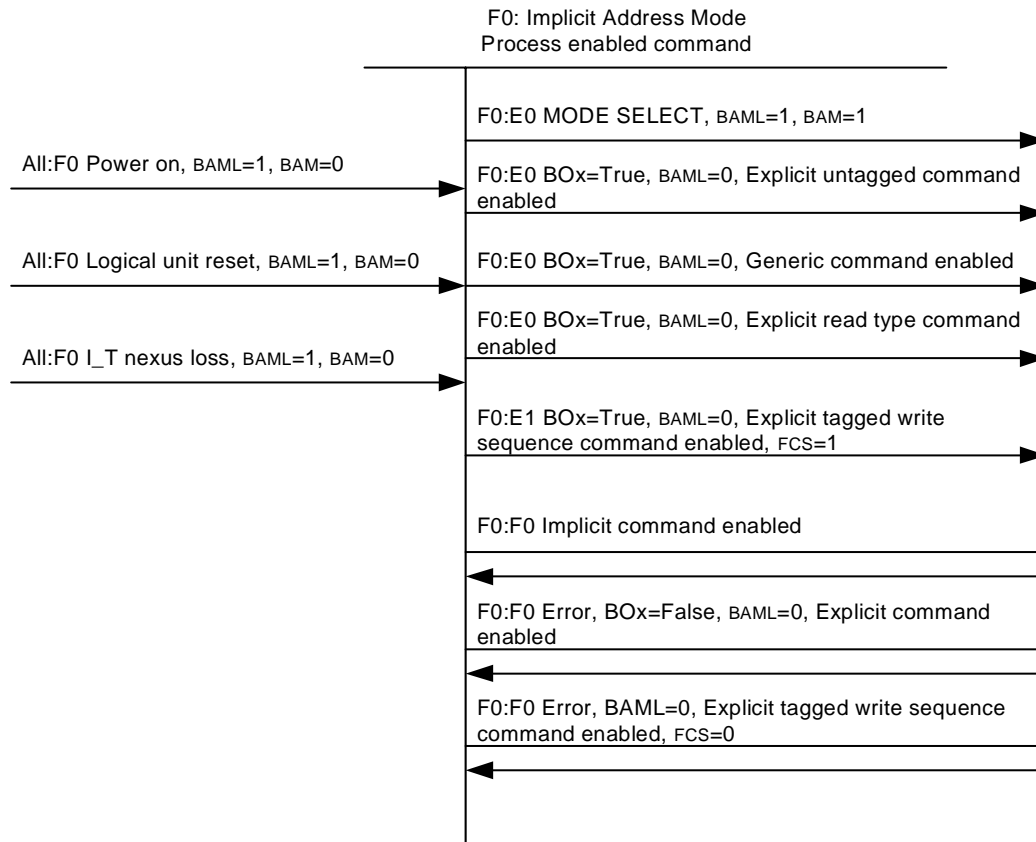
Figure 16 — Block address mode state diagram, Implicit Address Mode

State F0:Implicit Address Mode: This is the state for implicit address mode.

Transition All:F0: This transition shall occur when a power-on, logical unit reset, ot I_T nexus loss event occurs and the BAML bit is set to one and the BAM bit is set to zero.

Transition F0:E0: This transition shall occur when:

   a) a MODE SELECT command specifying a Device Configuration mode page with the BAML bit set to one and the BAM bit set to one completes with GOOD status;

   b) an explicit untagged command is enabled, the medium position is at BOx, and the BAML bit is set to zero;

   c) a generic command is enabled, the medium position is at BOx, and the BAML bit is set to zero; or

   d) an explicit read type command is enabled, the medium position is at BOx, and the BAML bit is set to zero.

Transition F0:E1: This transition shall occur when an explicit tagged write sequence command is enabled with the FCS bit set to one, the medium position is at BOx, and the BAML bit is set to zero.

Transition F0:F0: This transition shall occur when:

   a) an implicit command is enabled;

   b) an explicit command is enabled, the medium position is not at BOx, and the BAML bit is set to zero. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL

REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN IMPLICIT ADDRESS MODE; or

c) an explicit tagged write sequence command is enabled with the FCS bit set to zero. In this case the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to ILLEGAL COMMAND WHILE IN IMPLICIT ADDRESS MODE.

4.2.15 TapeAlert application client interface

4.2.15.1 Introduction to TapeAlert application client interface

TapeAlert information is accessed using the TapeAlert log page (see table 53) and configuration is performed using the Informational Exceptions Control mode page (see table 70). The MODE SENSE and MODE SELECT configuration of the TapeAlert interface is compatible with the Informational Exceptions Control mode page. The application client should first check the tape drive to determine whether it supports the TapeAlert log page. The default application client access to the TapeAlert log page is performed using a polling method, with the page control bits in the LOG SENSE command set to 00h.

The TapeAlert log page may be read at any time and should be read from the tape drive/autoloader device for the following:

a) at the beginning of a write/read application, even if media is not loaded;
b) immediately after an unrecoverable error during the write/read application;
c) at the end of each tape when the write/read application spans multiple tapes. If the tape is to be de-mounted then the log page should be read before the tape cartridge is de-mounted; or
d) at the end of a write/read application.

The application client may also poll the TapeAlert log page at regular intervals (e.g., every 60 seconds) while the tape drive is idle. The application client may use the Information Exceptions Control mode page to configure other access methods, depending on what options are supported by the tape drive.

Each time the application client reads the TapeAlert log page, it should check all 64 flags (see Table A.1) to discover which flags are active (there may be more than one). The definitions of the 64 flags are device type specific. There is one definition for tape drive/autoloader devices, and a different definition for stand-alone media changer devices (in libraries). For each flag active, the application client should communicate the defined error message and severity for that flag to the user and log it. If multiple flags are active simultaneously, they should be displayed together in ascending order of severity. At the beginning of each set of TapeAlert error messages, the tape device/library that initiated them should be identified. For the tape drive media-related flags (flags 4h, 7h, and Fh), the software label of the media should be included in the TapeAlert error messages so that the user is aware what piece of media the error refers to. Such information may also be displayed with the messages for other flags as well if required. The information read in the TapeAlert flags shall not in itself cause the application client to stop data transfer operations (e.g., a backup or restore operation).

4.2.15.2 TapeAlert log sense format

The TapeAlert interface to the tape drive is based on a LOG SENSE page (2Eh) containing 64 one-byte flags (see table A.1). The specific conditions for any specific flag to be active and inactive are vendor-specific.

The TapeAlert data is event based and the page control bits in the LOG SENSE command are not applicable and shall be ignored by the device server.

All flags shall be deactivated in the following circumstances:

a) after the TapeAlert log page is read. The TapeAlert flags shall be deactivated on a per-initiator basis such that active flags are available for other initiators;

  b)   when the specified corrective action has been taken (e.g., using a cleaning cartridge);
  c)   on logical unit reset; or
  d)   when the PCR field in the LOG SELECT command descriptor block is one (see SPC-3).

If a LOG SELECT command is enabled for the TapeAlert log page, the device server shall terminate the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.
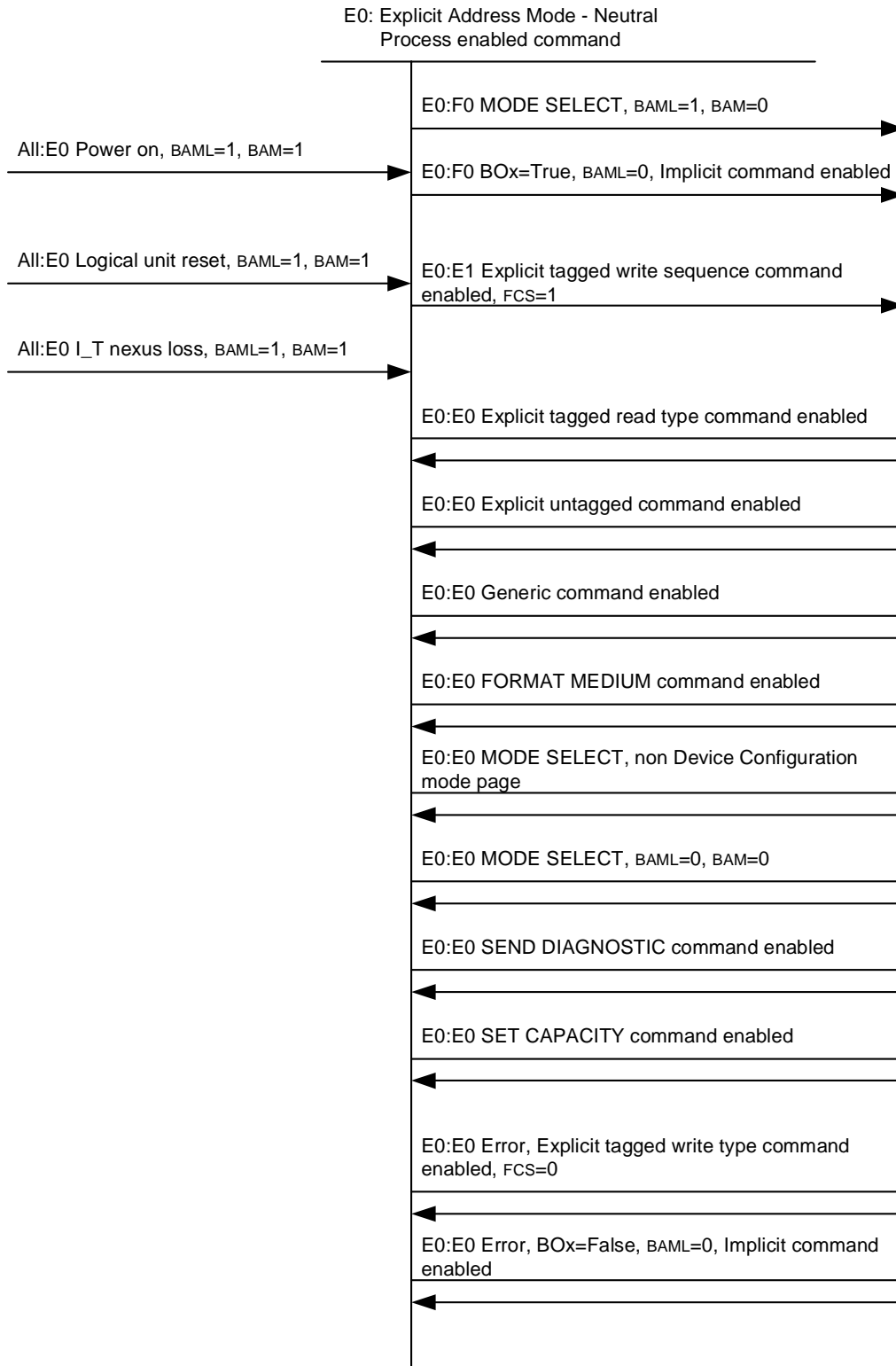
When a flag is deactivated by reading the TapeAlert log page, a flag shall not be active again until the error condition is removed (i.e., the specified corrective action has been taken). For example, if the cartridge in the drive is not data grade, once flag 8 has been deactivated by the application client reading the log page, it shall not be activated again until the cartridge has been removed. All other methods of deactivating the flag allow the flag to be activated again.

4.2.15.3 Tape drive/autoloader flag types and definitions

Table 6 specifies the three types of flags:

Table 6 — TapeAlert flag types

Severity	Urgent User Intervention	Risk of Data Loss	Explanation
Critical	X	X	
Warning		X	X
Information			X

Flag definitions are defined in table A.1 (see Annex A) for a tape drive or a tape drive with an attached media changer.

If the TapeAlert log page is supported, the flags described in table 7 shall be supported.

Table 7 — TapeAlert flags minimum subset

Flag Number	Flag Type	Definition
3h	Hard error	Active for any unrecoverable read/write/positioning error, and is internally deactivated when the media is de-mounted (this flag is active as specified in flag number 5h, or 6h).
4h	Media	Active for any unrecoverable read/write/positioning error that is due to faulty media, and is internally deactivated when the media is de-mounted.
5h	Read failure	Active for any unrecoverable read error where the diagnosis is uncertain and could either be faulty media or faulty drive hardware, and is internally deactivated when the media is de-mounted.

Table 7 — TapeAlert flags minimum subset (Continued)

Flag Number	Flag Type	Definition
6h	Write failure	Active for any unrecoverable write/positioning error where the diagnosis is uncertain and could either be faulty media or faulty drive hardware, and is internally deactivated when the media is de-mounted.
1Fh	Hardware B	Active when the tape drive fails its internal Power-On-Self-Tests (POST), and is not internally deactivated until the drive is powered off.

If the tape drive/autoloader does not support any of the TapeAlert flags not listed in table 7, then it is acceptable for unsupported flags to remain inactive.

The tape drive/autoloader flag definitions are grouped as shown in table 8:

Table 8 — TapeAlert flag definition grouping

Flag Number(s)	Definition
01h to 13h	Tape drive write/read management
14h to 19h	Cleaning management
1Ah to 27h	Tape drive hardware errors
28h to 31h	Tape autoloader errors
32h to 40h	Further tape errors

4.2.15.4 WORM TapeAlert flags

Two TapeAlert flags exist to support Write Once Read Many (WORM) media:

a) 3Bh, WORM Medium, Integrity Check Failed; and
b) 3Ch, WORM Medium, Overwrite Attempted.

If the device server supports TapeAlert flag 3Bh, it shall set that flag to one upon detecting that the integrity of the medium may be comprimised. If the device server supports TapeAlert flag 3C, it shall set that flag to one when an application client attempts to overwrite or erase user data on an archive tape (see 4.2.18).

In addition to the deactivation conditions for all TapeAlert flags (see 4.2.15.2), the device server shall set TapeAlert flags 3Bh and 3Ch to zero upon:

a) execution of a LOAD UNLOAD command with a load bit set to one (see 7.2) that results in a not ready to ready transition, or when both the medium and device server support MAM, that results in access to medium auxiliary memory only; or
b) execution of an autoload operation (see SPC-3) that results in a not ready to ready transition, or when both the medium and device server support MAM, that results in access to medium auxiliary memory only.

4.2.16 READ ATTRIBUTE and WRITE ATTRIBUTE command support

Support for the READ ATTRIBUTE and WRITE ATTRIBUTE commands (see SPC-3) is described in table 9 and table 10.

Table 9 — Device common attributes

ID	Attribute Name	Number of Bytes	Format
0002h	TAPEALERT FLAGS	8	Binary
0005h	ASSIGNING ORGANIZATION	8	ASCII
0006h	FORMATTED DENSITY CODE	1	Binary

The TAPEALERT FLAGS attribute provides a means of reporting the state of the TapeAlert flags for the previous load of the medium. Each TapeAlert flag occupies one bit (Flag 1 = MSB, Byte 1; Flag 64 = LSB, Byte 8). The bits specify all the TapeAlert flags that were set during the previous load, (i.e., the bits are "sticky" for the load).

The ASSIGNING ORGANIZATION attribute identifies the organization responsible for the specifications defining the values in the FORMATTED DENSITY CODE attribute. The ASSIGNING ORGANIZATION attribute should contain a vendor identification. The use of specific vendor identification other than the one associated with the device is allowed.

> NOTE 4 It is intended that the ASSIGNING ORGANIZATION attribute provide a unique vendor identification of the FORMATTED DENSITY CODE attribute. In the absence of a formal registration procedure, T10 maintains a list of known vendor identification codes for use in the Standard INQUIRY data (see SPC-3). Vendors are requested to voluntarily submit their identification codes to T10 to prevent duplication of codes (see SPC-3).

If the device server formats the medium using a format other than the one specified in the MEDIUM DENSITY CODE attribute (e.g., for compatibility with a previous generation format), then the FORMATTED DENSITY CODE attribute specifies the DENSITY CODE of the format chosen (see SPC-3). Otherwise this attribute shall be the same as the MEDIUM DENSITY CODE attribute.

Table 10 — Medium common attributes

ID	Attribute Name	Number of Bytes	Format
0402h	MEDIUM LENGTH	4	Binary
0403h	MEDIUM WIDTH	4	Binary
0404h	ASSIGNING ORGANIZATION	8	ASCII
0405h	MEDIUM DENSITY CODE	1	Binary

The MEDIUM LENGTH attribute specifies the length of the medium in meters. A value of 0h specifies that the length of the medium is undefined.

The MEDIUM WIDTH attribute specifies the width of the medium supported by this density. This attribute has units of tenths of millimeters. The value in this attribute shall be rounded up if the fractional value of the actual value is greater than or equal to 0,5. The MEDIUM WIDTH attribute may vary for a given density depending on the mounted medium. A value of 0h specifies the width of the medium is undefined.

The ASSIGNING ORGANIZATION attribute identifies the organization responsible for the specifications defining the values in the MEDIUM DENSITY CODE attribute. The ASSIGNING ORGANIZATION attribute should contain a vendor identification.

NOTE 5 It is intended that the ASSIGNING ORGANIZATION attribute provide a unique vendor identification of the MEDIUM DENSITY CODE attribute. In the absence of a formal registration procedure, T10 maintains a list of known vendor identification codes for use in the Standard INQUIRY data (see SPC-3). Vendors are requested to voluntarily submit their identification codes to T10 to prevent duplication of codes (see SPC-3).

4.2.17 Reservations

Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation. See SPC-3 for a description of reservations. The details of which commands are allowed under what types of reservations are described in table 11.

Commands from I_T_nexuses holding a reservation should complete normally. Table 11 specifies the behavior of commands from registered I_T_nexuses when a registrants only or all registrants persistent reservation is present.

NOTE 6 Due to the nature of streaming device types, Write Exclusive and Write Exclusive, Registrants Only modes of reservation do not protect an application client's continuity of operations when using the implicit address command set. While these modes do protect unauthorized modification of data, they do not protect from medium position changes that may result in errors due to incorrect position. It is the responsibility of the application client to manage this using means beyond the scope of this specification. Application clients should use exclusive modes of reservation while accessing the medium to prevent interference from other applications.

For each command, this standard and the SPC-3 standard define the conditions that result in RESERVATION CONFLICT.

Table 11 — SSC-3 commands that are allowed in the presence of various reservations

Command	Addressed LU 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
ERASE(6)	Conflict	Conflict	Allowed	Conflict	Conflict
ERASE(16)	Conflict	Conflict	Allowed	Conflict	Conflict
FORMAT MEDIUM	Conflict	Conflict	Allowed	Conflict	Conflict
LOAD UNLOAD	Conflict	Conflict	Allowed	Conflict	Conflict
LOCATE(10)	Allowed	Conflict	Allowed	Allowed	Conflict
LOCATE(16)	Allowed	Conflict	Allowed	Allowed	Conflict
READ(6)	Allowed	Conflict	Allowed	Allowed	Conflict
READ(16)	Allowed	Conflict	Allowed	Allowed	Conflict
READ BLOCK LIMITS	Allowed	Allowed	Allowed	Allowed	Allowed
READ POSITION	Allowed	Conflict	Allowed	Allowed	Conflict
READ REVERSE(6)	Allowed	Conflict	Allowed	Allowed	Conflict
READ REVERSE(16)	Allowed	Conflict	Allowed	Allowed	Conflict
RECOVER BUFFERED DATA	Allowed	Conflict	Allowed	Allowed	Conflict
REPORT DENSITY SUPPORT	Allowed	Allowed	Allowed	Allowed	Allowed

Table 11 — SSC-3 commands that are allowed in the presence of various reservations (Continued)

Command	Addressed LU 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**
REWIND	Allowed	Conflict	Allowed	Allowed	Conflict
SET CAPACITY	Conflict	Conflict	Allowed	Conflict	Conflict
SPACE(6)	Allowed	Conflict	Allowed	Allowed	Conflict
SPACE(16)	Allowed	Conflict	Allowed	Allowed	Conflict
VERIFY(6)	Allowed	Conflict	Allowed	Allowed	Conflict
VERIFY(16)	Allowed	Conflict	Allowed	Allowed	Conflict
WRITE(6)	Conflict	Conflict	Allowed	Conflict	Conflict
WRITE(16)	Conflict	Conflict	Allowed	Conflict	Conflict
WRITE FILEMARKS(6)	Conflict	Conflict	Allowed	Conflict	Conflict
WRITE FILEMARKS(16)	Conflict	Conflict	Allowed	Conflict	Conflict
Key: **LU**=Logical Unit, **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 terminate the command with RESERVATION CONFLICT status.					

4.2.18 Archive tape and WORM mode

4.2.18.1 WORM overview

The device server may optionally support a Write Once, Read Many (WORM) mode of operation. This mode of operation places additional restrictions on the processing of commands by the devices server.

4.2.18.2 Archive tape

An archive tape is identified by a format-defined method (e.g., the Medium Type attribute in the MAM data returning a Write Once Medium type value).

4.2.18.3 WORM mode

A device server that supports WORM mode and detects an archive tape is mounted shall enter WORM mode. While in WORM mode, WRITE, WRITE FILEMARKS, ERASE, FORMAT MEDIUM, SET CAPACITY, and MODE SELECT commands that alter the partitioning or format are subject to the restrictions applied to archive tape by the device server (see 8.3.7).

If a device server operating in WORM mode detects that one of these additional restrictions would be violated by a command being processed, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code shall be set to WORM MEDIUM - OVERWRITE ATTEMPTED. If a write protection is in effect for the device server or medium (see 4.2.11), it shall take precedence over the WORM mode violation.

If a device server operating in WORM mode is unable to determine if medium is such that one of these additional restrictions would be violated by a command being processed, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code shall be set to WORM MEDIUM - OVERWRITE ATTEMPTED. If a write protection is in effect for the device server or medium (see 4.2.11), it shall take precedence over the WORM mode violation.

If a device server that does not support WORM mode detects that an archive tape is mounted, it shall treat the medium as write protected. Any command that attempts to alter the medium shall be terminated with CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code shall be set to CANNOT WRITE MEDIUM - INCOMPATIBLE FORMAT.

5 Explicit address command descriptions for sequential-access devices

5.1 Summary of commands for explicit address mode

The explicit address command set for sequential-access devices shall be as shown in table 12. Commands specified as mandatory in table 12 shall be implemented if the explicit address command set is supported.

Refer to table 11 for a description of reservations.

The following command codes are vendor-specific: 02h, 06h, 07h, 09h, 0Ch, 0Dh, and 0Eh.

Table 12 — Explicit address command set for sequential-access devices

Command Name	Type	OpCode	Synchronize Operation Required[a]	Command Type	Reference
ACCESS CONTROL IN	O	86h	No	G	SPC-3
ACCESS CONTROL OUT	O	87h	No	G	SPC-3
CHANGE ALIAS	O	A4h/0Bh[c]	No	G	SPC-3
ERASE(16)	M	93h	Yes	W-E	5.2
EXTENDED COPY	O	83h	No	W or R[b]	SPC-3
FORMAT MEDIUM	O	04h	No	W	7.1
INQUIRY	M	12h	No	G-A	SPC-3
LOAD UNLOAD	O	1Bh	Yes	G	7.2
LOCATE(16)	M	92h	Yes	G-E	7.3
LOG SELECT	O	4Ch	No	G	SPC-3
LOG SENSE	O	4Dh	No	G-A	SPC-3
MODE SELECT(6)	M	15h	Yes[a]	W or R[b]	SPC-3
MODE SELECT(10)	O	55h	Yes[a]	W or R[b]	SPC-3
MODE SENSE(6)	M	1Ah	No	G	SPC-3
MODE SENSE(10)	O	5Ah	No	G	SPC-3

Key: M = Command implementation is mandatory.
    O = Command implementation is optional.
    X = See referenced standard or subclause.
    R = Read type command.
    W = Write type command.
    G = Generic type command.
    E = Explicit command.
    A = Allowed command while in write capable state.					

a. Refer to 4.2.8. b. This command has some specific actions that fall under write type commands and some that fall into read type commands. c. This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.					

Table 12 — Explicit address command set for sequential-access devices (Continued)

Command Name	Type	OpCode	Synchronize Operation Required[a]	Command Type	Reference
MOVE MEDIUM ATTACHED	O	A7h	Yes	G	SMC-2
PERSISTENT RESERVE IN	M	5Eh	No	G	SPC-3
PERSISTENT RESERVE OUT	M	5Fh	No	G	SPC-3
PREVENT ALLOW MEDIUM REMOVAL	O	1Eh	No	G-A	SPC-3
READ(16)	M	88h	Yes	R-E	5.3
READ ATTRIBUTES	O	8Ch	No	G	SPC-3
READ BLOCK LIMITS	M	05h	No	G-A	7.4
READ BUFFER	O	3Ch	Yes	G	SPC-3
READ ELEMENT STATUS ATTACHED	O	B4h	No	G-A	SMC-2
READ POSITION	M	34h	No	G-A	7.5
READ REVERSE(16)	O	81h	Yes	R-E	5.4
RECEIVE COPY RESULTS	O	84h	No	G	SPC-3
RECEIVE DIAGNOSTIC RESULTS	O	1Ch	No	G	SPC-3
RECOVER BUFFERED DATA	O	14h	May	R	7.6
REPORT ALIAS	O	A3h/0Bh[c]	No	G	SPC-3
REPORT DENSITY SUPPORT	M	44h	No	G-A	7.7
REPORT DEVICE IDENTIFIER	O	A3h/05h[c]	No	G	SPC-3
REPORT LUNS	X	A0h	No	G-A	SPC-3
REPORT SUPPORTED OPERATION CODES	O	A3h/0Ch[c]	No	G	SPC-3
REQUEST SENSE	M	03h	No	G	SPC-3
REPORT TARGET PORT GROUPS	O	A3h/0Ah[c]	No	G	SPC-3
REWIND	M	01h	Yes	G	7.8
SEND DIAGNOSTIC	M	1Dh	Yes[a]	W or R[b]	SPC-3

Key: M = Command implementation is mandatory.
    O = Command implementation is optional.
    X = See referenced standard or subclause.
    R = Read type command.
    W = Write type command.
    G = Generic type command.
    E = Explicit command.
    A = Allowed command while in write capable state.

a. Refer to 4.2.8.
b. This command has some specific actions that fall under write type commands and some that fall into read type commands.
c. This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.

Table 12 — Explicit address command set for sequential-access devices (Continued)

Command Name	Type	OpCode	Synchronize Operation Required[a]	Command Type	Reference
SET CAPACITY	O	0Bh	May	W	7.9
SET DEVICE IDENTIFIER	O	A4h/06h[c]	No	G	SPC-3
SET TARGET PORT GROUPS	O	A4h/0Ah[c]	No	G	SPC-3
SPACE(16)	O	91h	May	G-E	7.10
TEST UNIT READY	M	00h	No	G	SPC-3
VERIFY(16)	O	8Fh	Yes	R-E	5.5
WRITE(16)	M	8Ah	No	W-E	5.6
WRITE ATTRIBUTES	O	8Dh	No	G	SPC-3
WRITE BUFFER	O	3Bh	Yes[a]	G	SPC-3
WRITE FILEMARKS(16)	M	80h	May	W-E	5.7

Key: M = Command implementation is mandatory.
O = Command implementation is optional.
X = See referenced standard or subclause.
R = Read type command.
W = Write type command.
G = Generic type command.
E = Explicit command.
A = Allowed command while in write capable state.

a. Refer to 4.2.8.
b. This command has some specific actions that fall under write type commands and some that fall into read type commands.
c. This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.

5.2 ERASE(16) command

The ERASE(16) command (see table 13) causes part or all of the medium to be erased beginning at the logical object identifier and partition specified in the command descriptor block. Erased means the medium shall be

erased or a pattern shall be written on the medium that appears as a gap. Prior to performing the erase operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 13 — ERASE(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (93h)							
1	Reserved				FCS	LCS	IMMED	LONG
2	Reserved							
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	CONTROL							

A first command in sequence (FCS) bit of one specifies this command is the first command in a tagged write sequence. An FCS bit of zero specifies this command is not the first command in a tagged write sequence.

A last command in sequence (LCS) bit of one specifies this command is the last command in a tagged write sequence. An LCS bit of zero specifies this command is not the last command in a tagged write sequence.

An immediate (IMMED) bit of zero specifies the device server shall not return status until the erase operation has completed. Interpretation of an IMMED bit of one depends on the value of the LONG bit, see below. However, for all values of the LONG bit, if CHECK CONDITION status is returned for an ERASE(16) command with an IMMED bit of one, the erase operation shall not be performed.

A LONG bit of one specifies all remaining medium in the current partition shall be erased beginning at the specified logical object identifier and partition. If the IMMED bit is one, the device server shall return status as soon as all buffered logical objects have been written to the medium and the command descriptor block of the ERASE(16) command has been validated. The logical position following an ERASE(16) command with a LONG bit of one is not specified by this standard.

> NOTE 7 Some logical units may reject an ERASE(16) command with the LONG bit set to one if the logical object identifier is not zero.

A LONG bit of zero specifies a vendor-specific erase gap shall be written to the medium. The logical position following a ERASE(16) command with a LONG bit of zero shall be at the specified logical object identifier and partition. If the IMMED bit is one, the device server shall return status as soon as the command descriptor block has

been validated. Erase gaps may be used in application client controlled error recovery or update in place applications.

Editors Note 1 - DAP: Since the gap size field has been obsoleted the ability for application client controlled error recovery or update in place applications is suspect. Suggest removing the sentence.

If the logical unit encounters early-warning during an ERASE(16) command, and any buffered logical objects remain to be written, the device server action shall be as defined for the early-warning condition of the WRITE(16) command (see 5.6). If the LONG bit is zero, the erase operation shall terminate with CHECK CONDITION status and the sense data shall be set as defined for the WRITE(16) command. Any count of pending buffered erases shall not be reported as part of the value returned in the INFORMATION field or in the READ POSITION response data.

The PARTITION and LOGICAL OBJECT IDENTIFIER fields specify the position at which the ERASE(16) command shall start. If the current position does not match the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the erase operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. The logical position is undefined following a locate operation failure with a LONG bit of zero.

5.3 READ(16) command

The READ(16) command (see table 14) requests that the device server transfer one or more logical block(s) to the application client beginning at the logical object identifier and partition specified in the command descriptor block. Prior to performing the read operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 14 — READ(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (88h)							
1	Reserved						SILI	FIXED
2	Reserved							
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	(MSB)							
13				TRANSFER LENGTH				
14								(LSB)
15	CONTROL							

The FIXED bit specifies whether fixed-block transfers or variable-block transfers are to be used. Refer to the READ BLOCK LIMITS command (see 7.4) for additional information about fixed-block transfers and variable-block transfers.

If the FIXED bit is one, the TRANSFER LENGTH specifies the number of fixed-length blocks to be transferred, using the current block length reported in the mode parameters block descriptor (see SPC-3). If the FIXED bit is zero, a variable-length block is requested with the TRANSFER LENGTH specifying the maximum number of bytes allocated for the returned logical block.

A successful READ(16) command with a FIXED bit of one shall transfer the requested transfer length times the current block length in bytes to the application client. A successful READ(16) command with a FIXED bit of zero shall transfer the requested transfer length in bytes to the application client. Upon completion, the logical position shall be after the last logical block transferred (end-of-partition side).

If the suppress incorrect-length indicator (SILI) bit is one and the FIXED bit is zero, the device server shall:

    a) report CHECK CONDITION status for an incorrect-length condition only if the overlength condition exists and the BLOCK LENGTH field in the mode parameter block descriptor is nonzero (see SPC-3); or
    b) not report CHECK CONDITION status if the only error is the underlength condition, or if the only error is the overlength condition and the BLOCK LENGTH field of the mode parameters block descriptor is zero.

NOTE 8 Since the residual information normally provided in the INFORMATION field of the sense data may not be available when the SILI bit is set to one, other methods for determining the actual block length should be used (e.g., including length information in the logical block).

If the SILI bit is one and the FIXED bit is one, the device server shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

If the SILI bit is zero and an incorrect-length logical block is read, CHECK CONDITION status shall be returned and the ILI and VALID bits shall be set to one in the sense data with an additional sense code of NO ADDITIONAL SENSE INFORMATION. Upon termination, the logical position shall be after the incorrect-length logical block (end-of-partition side). If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read (not including the incorrect-length logical block). If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length minus the actual logical block length. Logical units that do not support negative values shall set the INFORMATION field to zero if the overlength condition exists.

NOTE 9 In the above case with the FIXED bit of one, only the position of the incorrect-length logical block may be determined from the sense data. The actual length of the incorrect logical block is not reported. Other means may be used to determine its actual length (e.g., read it again with the fixed bit set to zero).

The LOGICAL OBJECT IDENTIFIER and PARTITION fields specify the position at which the READ(16) command shall start. If the TRANSFER LENGTH field is not set to zero and the current logical position does not match the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the read operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. The INFORMATION field in the sense data shall be set to the requested transfer length. Following a locate operation failure the logical position is undefined.

If the TRANSFER LENGTH field is set to zero, no data shall be transferred and the current logical position shall not be changed. This condition shall not be considered an error.

In the case of an unrecovered read error, if the FIXED bit is one, the sense data VALID bit shall be set to one and the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read (not including the unrecovered logical blocks). If the FIXED bit is zero, the sense data VALID bit shall be set to one and the INFORMATION field shall be set to the requested transfer length. Upon termination, the logical position shall be after the unrecovered logical block.

If the device server encounters a filemark during a READ(16) command, CHECK CONDITION status shall be returned and the FILEMARK and VALID bits shall be set to one in the sense data. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate, and the additional sense code shall be set to FILEMARK DETECTED. Upon termination, the logical position shall be after the filemark (end-of-partition side). If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

If the device server encounters early-warning during a READ(16) command and the REW bit is set to one in the Device Configuration mode page (see 8.3.3), CHECK CONDITION status shall be returned upon completion of the current logical block. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate, and the additional sense code shall be set to END-OF-PARTITON/MEDIUM DETECTED. The EOM and VALID bits shall be set to one in the sense data. Upon termination, the logical position shall be after the last logical block transferred (end-of-partition side). If the FIXED bit is one, the INFORMATION field in the sense data shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field in the sense data shall be set to the requested transfer length minus the actual logical block length. The device server shall not return CHECK CONDITION status when early-warning is encountered if the REW bit is zero.

NOTE 10 A REW bit of one is not recommended for most applications since read data may be present after early-warning.

If the device server encounters end-of-data during a READ(16) command, CHECK CONDITION status shall be returned, the sense key shall be set to BLANK CHECK, and the VALID bit shall be set to one in the sense data. If end-of-data is encountered at or after early-warning, the EOM bit shall also be set to one. Upon termination, the logical position shall be immediately after the last recorded logical object (end-of-partition side). If the FIXED bit is one, the INFORMATION field in the sense data shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

If the device server encounters end-of-partition during a READ(16) command, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, and the EOM and VALID bits shall be set to one in the sense data. The medium position following this condition is not defined. If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field in the sense data shall be set to the requested transfer length.

NOTE 11 If a READ(16) command terminates with an error condition other than ILLEGAL REQUEST, and no data transfer has occurred, the logical position of the medium is undefined. The application client should issue a READ POSITION(16) command to determine the logical position.

5.4 READ REVERSE(16) command

The READ REVERSE(16) command (see table 15) requests that the device server transfer one or more logical block(s) to the application client beginning at the logical object identifier and partition specified in the command descriptor block. Prior to performing the read reverse operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 15 — READ REVERSE(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (81h)							
1	Reserved					BYTORD	SILI	FIXED
2	Reserved							
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	(MSB)							
13				TRANSFER LENGTH				
14								(LSB)
15	CONTROL							

This command is similar to the READ(16) command except that medium motion is in the reverse direction. Upon completion of a READ REVERSE(16) command, the logical position shall be before the last logical block transferred (beginning-of-partition side).

A byte order (BYTORD) bit of zero specifies all logical block(s), and the byte(s) within the logical block(s), are transferred in the reverse order. The order of bits within each byte shall not be changed. A BYTORD bit of one specifies all logical block(s) are transferred in the reverse order but the byte(s) within the logical block(s) are transferred in the same order as returned by the READ(16) command. Support for either value of the BYTORD bit is optional.

Refer to the READ(16) command (see 5.3) for a description of the FIXED bit, the SILI bit, the TRANSFER LENGTH field, and any conditions that may result from incorrect usage of these fields.

The LOGICAL OBJECT IDENTIFIER and PARTITION fields specify the position at which the READ REVERSE(16) command shall start. If the TRANSFER LENGTH field is not set to zero and the current logical position does not match the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the read reverse operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. The INFORMATION field in the sense data shall be set to the requested transfer length. Following a locate operation failure the logical position is undefined.

Filemarks, incorrect-length logical blocks, and unrecovered read errors are handled the same as in the READ(16) command, except that upon termination the logical position shall be before the filemark, incorrect-length logical block, or unrecovered logical block (beginning-of-partition side).

If the device server encounters beginning-of-partition during a READ REVERSE(16) command, CHECK CONDITION status shall be returned and the EOM and VALID bits shall be set to one in the sense data. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks transferred. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

5.5 VERIFY(16) command

The VERIFY(16) command (see table 16) requests that the device server verify one or more logical block(s) beginning at the logical object identifier and partition specified in the command descriptor block. Prior to performing the verify operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 16 — VERIFY(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Fh)							
1	Reserved					IMMED	BYTCMP	FIXED
2	Reserved							
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	(MSB)							
13				VERIFICATION LENGTH				
14								(LSB)
15	CONTROL							

An immediate (IMMED) bit of zero specifies the command shall not return status until the verify operation has completed. An IMMED bit of one specifies status shall be returned as soon as the command descriptor block has been validated; but after all verification data has been transferred from the application client to the device server, if the BYTCMP bit is one.

> NOTE 12 In order to ensure that no errors are lost, the application client should set the IMMED bit to zero on the last VERIFY(16) command when issuing a series of VERIFY(16) commands.

A byte compare (BYTCMP) bit of zero specifies the verification shall be simply a medium verification (e.g., CRC, ECC). No data shall be transferred from the application client to the device server.

A BYTCMP bit of one specifies the device server shall perform a byte-by-byte compare of the data on the medium and the data transferred from the application client. Data shall be transferred from the application client to the device server as in a WRITE(16) command (see 5.6). If the BYTCMP bit is one and the byte compare option is not supported, the device server shall terminate the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

The LOGICAL OBJECT IDENTIFIER and PARTITION fields specify the position where the VERIFY(16) command shall start. If the VERFICATION LENGTH field is not zero and the current logical position does not match the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the verify operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. The INFORMATION field in the sense data shall be set to the requested verification length. Following a locate operation failure the logical position is undefined.

The VERIFICATION LENGTH field specifies the amount of data to verify, in logical blocks or bytes, as specified by the FIXED bit. Refer to the READ(16) command (see 5.3) for a description of the FIXED bit and any error conditions that may result from incorrect usage. If the VERIFICATION LENGTH field is zero, no data shall be verified and the current logical position shall not be changed. This condition shall not be considered an error.

The VERIFY(16) command shall terminate as follows:

  a) when the verification length has been satisfied;
  b) when an incorrect-length logical block is encountered;
  c) when a filemark is encountered;
  d) when end-of-data is encountered;
  e) when the end-of-partition is encountered;
  f) when early-warning is encountered (if the REW bit is one in the Device Configuration mode page, see 8.3.3); or
  g) when an unrecoverable read error is encountered.

The status and sense data for each of the termination conditions are handled in the same manner as in the READ(16) command (see 5.3). Upon successful completion of a VERIFY(16) command, the logical position shall be after the last logical block verified (end-of-partition side).

If the data does not compare (BYTCMP bit of one), the command shall terminate with CHECK CONDITION status, the sense data VALID bit shall be set to one, the sense key shall be set to MISCOMPARE, and the additional sense code shall be set to MISCOMPARE DURING VERIFY OPERATION. If the FIXED bit is one, the INFORMATION field shall be set to the requested verification length minus the actual number of logical blocks successfully verified. If the FIXED bit is zero, the INFORMATION field shall be set to the requested verification length minus the actual number of bytes successfully verified. This number may be larger than the requested verification length if the error occurred on a previous VERIFY(16) command with an IMMED bit of one. Upon termination, the medium shall be positioned after the logical block containing the miscompare (end-of-partition side).

5.6 WRITE(16) command

The WRITE(16) command (see table 17) requests that the device server write the logical block that is transferred from the application client to the logical object identifier and partition specified in the command descriptor block.

Table 17 — WRITE(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Ah)							
1	Reserved				FCS	LCS	Rsvd	FIXED
2	Reserved							
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	(MSB)							
13				TRANSFER LENGTH				
14								(LSB)
15	CONTROL							

A first command in sequence (FCS) bit of one specifies this command is the first command in a tagged write sequence. An FCS bit of zero specifies this command is not the first command in a tagged write sequence.

A last command in sequence (LCS) bit of one specifies this command is the last command in a tagged write sequence. An LCS bit of zero specifies this command is not the last command in a tagged write sequence.

The FIXED bit specifies whether fixed-block transfers or variable-block tansfers are to be used. See the READ BLOCK LIMITS command (see 7.4) for additional information about fixed-block transfers and variable-block transfers.

If the FIXED bit is one, the TRANSFER LENGTH value specifies the number of fixed-length blocks to be transferred, using the current block length reported in the mode parameter block descriptor (see 8.3). If the FIXED bit is zero, a single logical block is transferred with TRANSFER LENGTH specifying the logical block length in bytes.

The LOGICAL OBJECT IDENTIFIER and PARTITION fields specify the position where the WRITE(16) command shall start. If the TRANSFER LENGTH field is not set to zero and the current logical position does not match the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the write operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. The INFORMATION field in the sense data shall be set to the requested transfer length. Following a locate operation failure the logical position is undefined.

If the TRANSFER LENGTH field is zero, no data shall be transferred and the current logical position shall not be changed. This condition shall not be considered an error.

A WRITE(16) command may be buffered or unbuffered, as specified by the BUFFERED MODE field of the mode parameter header (see 8.3). When operating in unbuffered mode (see 3.1.70), the device server shall not return GOOD status until all logical block(s) are successfully written to the medium. When operating in buffered mode (see 3.1.8), the device server may return GOOD status as soon as all logical block(s) are successfully transferred to the logical unit's object buffer.

> NOTE 13 For compatibility with devices implemented prior to this version of this International Standard, a WRITE FILEMARKS(16) command with the IMMED bit set to zero should be issued when completing a buffered write operation to perform a synchronize operation (see 4.2.8).

If the device server enables a WRITE(16) command while positioned between EW and EOP, or encounters EW during the processing of a WRITE(16) command, an attempt to finish writing any data may be made as determined by the current settings of the REW and SEW bits in the Device Configuration mode page (see 8.3.3). The command shall terminate with CHECK CONDITION status and the additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED. If all data that is to be written is successfully transferred to the medium, the sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If the device server is unable to transfer any data, buffered or unbuffered, when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW. If the SEW bit is set to zero, the EOM bit shall be set to one in the sense data. If the SEW bit is set to one, the EOM and VALID bits shall be set to one in the sense data.

The INFORMATION field shall be set as follows:

a) if the FIXED bit is set to one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks transferred to the device server; or
b) if the FIXED bit is set to zero, the INFORMATION field shall be set to the requested transfer length.

The device server should perform a synchronize operation (see 4.2.8) after the first early-warning indication has been returned to the application client (see 4.2.3).

> NOTE 14 For some application clients it is important to recognize an error if end-of-partition is encountered during the processing of a WRITE(16) command, without regard for whether all data that is to be written is successfully transferred to the medium. The VOLUME OVERFLOW sense key may always validly be returned if end-of-partition is encountered while writing, and such usage is recommended. Reporting the MEDIUM ERROR sense key may cause confusion as to whether there was really defective medium encountered during the processing of the last WRITE(16) command.

If a WRITE(16) command is terminated early, an incomplete logical block (i.e., a logical block not completely transferred to the device server from the initiator) shall be discarded. The incomplete logical block may be accessible prior to new data being written to the media. The device server shall be logically positioned after the last logical block that was successfully transferred.

5.7 WRITE FILEMARKS(16) command

The WRITE FILEMARKS(16) command (see table 18) requests that the device server write the specified number of filemarks to the logical object identifier and partition specified in the command descriptor block.

Table 18 — WRITE FILEMARKS(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (80h)							
1	Reserved				FCS	LCS	Obsolete	IMMED
2	Reserved							
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	(MSB)							
13				TRANSFER LENGTH				
14								(LSB)
15	CONTROL							

A first command in sequence (FCS) bit of one specifies this command is the first command in a tagged write sequence. An FCS bit of zero specifies this command is not the first command in a tagged write sequence.

A last command in sequence (LCS) bit of one specifies this command is the last command in a tagged write sequence. An LCS bit of zero specifies this command is not the last command in a tagged write sequence.

An immediate (IMMED) bit of one specifies the device server shall return status as soon as the command descriptor block has been validated. An IMMED bit of one is only valid if the device is operating in buffered mode (see 3.1.8). If the IMMED bit is set to one and the device is operating in unbuffered mode the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

An IMMED bit of zero specifies the device server shall not return status until the write operation has completed. Any buffered logical objects shall be written to the medium prior to completing the command.

> NOTE 15 Upon completion of any buffered write operation, the application client may issue a WRITE FILEMARKS(16) command with the IMMED bit set to zero and the TRANSFER LENGTH field set to zero to perform a synchronize operation (see 4.2.8).

The LOGICAL OBJECT IDENTIFIER and PARTITION fields specify the position where the WRITE FILEMARKS(16) command shall start. If the TRANSFER LENGTH field is not set to zero and the current logical position does not match

the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the write filemarks operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. The INFORMATION field in the sense data shall be set to the requested transfer length. Following a locate operation failure the logical position is undefined.

If the device server enables a WRITE FILEMARKS(16) command while positioned between EW and EOP, or encounters EW during the processing of a WRITE FILEMARKS(16) command, an attempt to finish writing any buffered logical objects may be made, as determined by the current settings of the REW and SEW bits in the Device Configuration mode page (see 8.3.3). The command shall terminate with CHECK CONDITION status and the additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED. If all buffered logical objects to be written are successfully transferred to the medium, the sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If the device server is unable to transfer any buffered logical object when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW. If the SEW bit is set to zero, the EOM bit shall be set to one in the sense data. If the SEW bit is set to one, the EOM and VALID bits shall be set to one in the sense data.

The INFORMATION field shall be set to the requested transfer length minus the actual number of filemarks:

   a)   written to the object buffer if the IMMED bit is set to one; or
   b)   written to the medium if the IMMED bit is set to zero.

The device server should perform a synchronize operation (see 4.2.8) after the first early-warning indication has been returned to the application client (see 4.2.3).

6 Implicit address command descriptions for sequential-access devices

6.1 Summary of commands for implicit address mode

The implicit address commands for sequential-access devices are shown in table 19. Commands specified as mandatory in table 19 shall be implemented if the implicit address command set is supported.

If a synchronize operation is required for a command, the synchronize operation shall be performed as specified in 4.2.8.

Refer to table 11 for a description of reservations.

The following command codes are vendor-specific: 02h, 06h, 07h, 09h, 0Ch, 0Dh, and 0Eh.

Table 19 — Implicit address command set for sequential-access devices

Command Name	Type	OpCode	Synchronize Operation Required[a]	Reference
ACCESS CONTROL IN	O	86h	No	SPC-3
ACCESS CONTROL OUT	O	87h	No	SPC-3
CHANGE ALIAS	O	A4h/0Bh[b]	No	SPC-3
ERASE(6)	M	19h	Yes	6.2
EXTENDED COPY	O	83h	No	SPC-3
FORMAT MEDIUM	O	04h	No	7.1
INQUIRY	M	12h	No	SPC-3
LOAD UNLOAD	O	1Bh	Yes	7.2
LOCATE(10)	M	2Bh	Yes	6.3
LOCATE(16)	M	92h	Yes	7.3
LOG SELECT	O	4Ch	No	SPC-3
LOG SENSE	O	4Dh	No	SPC-3
MODE SELECT(6)	M	15h	Yes[a]	SPC-3
MODE SELECT(10)	O	55h	Yes[a]	SPC-3
MODE SENSE(6)	M	1Ah	No	SPC-3
MODE SENSE(10)	O	5Ah	No	SPC-3
MOVE MEDIUM ATTACHED	O	A7h	Yes	SMC-2
PERSISTENT RESERVE IN	M	5Eh	No	SPC-3
Key: M = Command implementation is mandatory. O = Command implementation is optional. X = See referenced standard or subclause.				
a. Refer to 4.2.8. b. This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.				

Table 19 — Implicit address command set for sequential-access devices (Continued)

Command Name	Type	OpCode	Synchronize Operation Required[a]	Reference
PERSISTENT RESERVE OUT	M	5Fh	No	SPC-3
PREVENT ALLOW MEDIUM REMOVAL	O	1Eh	No	SPC-3
READ(6)	M	08h	Yes	6.4
READ ATTRIBUTES	O	8Ch	No	SPC-3
READ BLOCK LIMITS	M	05h	No	7.4
READ BUFFER	O	3Ch	Yes	SPC-3
READ ELEMENT STATUS ATTACHED	O	B4h	No	SMC-2
READ POSITION	M	34h	No	7.5
READ REVERSE(6)	O	0Fh	Yes	6.5
RECEIVE COPY RESULTS	O	84h	No	SPC-3
RECEIVE DIAGNOSTIC RESULTS	O	1Ch	No	SPC-3
RECOVER BUFFERED DATA	O	14h	May	7.6
REPORT DENSITY SUPPORT	M	44h	No	7.7
REPORT ALIAS	O	A3h/0Bh[b]	No	SPC-3
REPORT DEVICE IDENTIFIER	O	A3h/05h[b]	No	SPC-3
REPORT LUNS	X	A0h	No	SPC-3
REPORT SUPPORTED OPERATION CODES	O	A3h/0Ch[b]	No	SPC-3
REPORT TARGET PORT GROUPS	O	A3h/0Ah[b]	No	SPC-3
REQUEST SENSE	M	03h	No	SPC-3
REWIND	M	01h	Yes	7.8
SEND DIAGNOSTIC	M	1Dh	Yes[a]	SPC-3
SET CAPACITY	O	0Bh	May	7.9
SET DEVICE IDENTIFIER	O	A4h/06h[b]	No	SPC-3
SET TARGET PORT GROUPS	O	A4h/0Ah[b]	No	SPC-3
SPACE(6)	M	11h	May	6.6
SPACE(16)	O	91h	May	7.10
TEST UNIT READY	M	00h	No	SPC-3
VERIFY(6)	O	13h	Yes	6.7

Key: M = Command implementation is mandatory.
    O = Command implementation is optional.
    X = See referenced standard or subclause.				

a. Refer to 4.2.8. b. This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.				

Table 19 — Implicit address command set for sequential-access devices (Continued)

Command Name	Type	OpCode	Synchronize Operation Required[a]	Reference
WRITE(6)	M	0Ah	No	6.8
WRITE ATTRIBUTES	O	8Dh	No	SPC-3
WRITE BUFFER	O	3Bh	Yes[a]	SPC-3
WRITE FILEMARKS(6)	M	10h	May	6.9

Key: M = Command implementation is mandatory.
    O = Command implementation is optional.
    X = See referenced standard or subclause.				

a. Refer to 4.2.8. b. This command is defined by a combination of operation code and service action. The operation code value is shown preceding the slash and the service action value is shown after the slash.				

6.2 ERASE(6) command

The ERASE(6) command (see table 20) causes part or all of the medium to be erased beginning at the current position. Erased means the medium shall be erased or a pattern shall be written on the medium that appears as a gap. Prior to performing the erase operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 20 — ERASE(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (19h)							
1	Reserved						IMMED	LONG
2	Reserved							
3	Reserved							
4	Reserved							
5	CONTROL							

An immediate (IMMED) bit of zero specifies the device server shall not return status until the erase operation has completed. Interpretation of an IMMED bit of one depends on the value of the LONG bit, see below. However, for all values of the LONG bit, if CHECK CONDITION status is returned for an ERASE(6) command with an IMMED bit of one, the erase operation shall not be performed.

A LONG bit of one specifies all remaining medium in the current partition shall be erased beginning at the current logical position. If the IMMED bit is one, the device server shall return status as soon as all buffered logical objects have been written to the medium and the command descriptor block of the ERASE(6) command has been validated. The logical position following an ERASE(6) command with a LONG bit of one is not specified by this standard.

> NOTE 16 Some logical units may reject an ERASE(6) command with the LONG bit set to one if the logical unit is not at beginning-of-partition.

A LONG bit of zero specifies a vendor-specific erase gap shall be written to the medium. If the IMMED bit is one, the device server shall return status as soon as the command descriptor block has been validated. Erase gaps may be used in application client controlled error recovery or update in place applications.

If the logical unit encounters early-warning during an ERASE(6) command, and any buffered logical objects remain to be written, the device server action shall be as defined for the early-warning condition of the WRITE(6) command (see 6.8). If the LONG bit is zero, the erase operation shall terminate with CHECK CONDITION status and set the sense data as defined for the WRITE(6) command. Any count of pending buffered erases shall not be reported as part of the value returned in the INFORMATION field or in the READ POSITION response data.

6.3 LOCATE(10) command

The LOCATE(10) command (see table 21) causes the logical unit to position the medium to the specified logical object with a matching logical object identifier in the specified partition. Upon completion, the logical position shall

be before the specified logical object. Prior to performing the locate operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 21 — LOCATE(10) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Bh)							
1	Reserved					BT	CP	IMMED
2	Reserved							
3	(MSB)							
4	LOGICAL OBJECT IDENTIFIER							
5								
6								(LSB)
7	Reserved							
8	PARTITION							
9	CONTROL							

A block identifier type (BT) bit of one specifies the value in the LOGICAL OBJECT IDENTIFIER field shall be interpreted as a vendor-specific value. A BT bit of zero specifies the value in the LOGICAL OBJECT IDENTIFIER field shall be interpreted as a logical object identifier (see 3.1.40).

A change partition (CP) bit of one specifies a change to the partition specified in the PARTITION field shall occur prior to positioning to the logical object specified in the LOGICAL OBJECT IDENTIFIER field. A CP bit of zero specifies no partition change shall occur and the PARTITION field shall be ignored.

An immediate (IMMED) bit of zero specifies the device server shall not return status until the locate operation has completed. If the IMMED bit is one, the device server shall return status as soon as all buffered logical objects have been written to the medium and the command descriptor block of the LOCATE(10) command has been validated. If CHECK CONDITION status is returned for a LOCATE(10) command with an IMMED bit of one, the locate operation shall not be performed.

The LOGICAL OBJECT IDENTIFIER field specifies the logical object identifier to which the logical unit shall position the medium based on the current setting of the BT bit. An otherwise valid LOCATE(10) command to any position between beginning-of-data and the position immediately after the last block in the partition (position at end-of-data) shall not return a sense key of ILLEGAL REQUEST. A LOCATE(10) to a position past end-of-data shall return CHECK CONDITION status and the sense key shall be set to BLANK CHECK. Additionally, the sense data EOM bit shall be set to one if end-of-data is located at or after early-warning.

If end-of-partition is encountered, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, and the sense data EOM bit shall be set to one.

The PARTITION field specifies the partition to select if the CP bit is one. Refer to the sequential-access device model (see 4.2.4) and the Medium Partition mode page (see 8.3.4) for additional information about partitioning.

The logical unit position is undefined if a LOCATE(10) command fails with a sense key other than ILLEGAL REQUEST.

6.4 READ(6) command

The READ(6) command (see table 22) requests that the device server transfer one or more logical block(s) to the application client beginning with the next logical block. Prior to performing the read operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 22 — READ(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (08h)							
1	Reserved						SILI	FIXED
2	(MSB)							
3				TRANSFER LENGTH				
4								(LSB)
5	CONTROL							

The FIXED bit specifies whether fixed-block transfers or variable-block transfers are to be used. Refer to the READ BLOCK LIMITS command (see 7.4) for additional information about fixed-block transfers and variable-block transfers.

If the FIXED bit is one, the TRANSFER LENGTH specifies the number of fixed-length blocks to be transferred, using the current block length reported in the mode parameters block descriptor (see 8.3). If the FIXED bit is zero, a variable-length block is requested with the TRANSFER LENGTH specifying the maximum number of bytes allocated for the returned logical block.

A successful READ(6) command with a FIXED bit of one shall transfer the requested transfer length times the current block length in bytes to the application client. A successful READ(6) command with a FIXED bit of zero shall transfer the requested transfer length in bytes to the application client. Upon completion, the logical position shall be after the last logical block transferred (end-of-partition side).

If the suppress incorrect-length indicator (SILI) bit is one and the FIXED bit is zero, the device server shall:

a) report CHECK CONDITION status for an incorrect-length condition only if the overlength condition exists and the BLOCK LENGTH field in the mode parameter block descriptor is nonzero (see 8.3); or
b) not report CHECK CONDITION status if the only error is the underlength condition, or if the only error is the overlength condition and the BLOCK LENGTH field of the mode parameters block descriptor is zero.

> NOTE 17 Since the residual information normally provided in the INFORMATION field of the sense data may not be available when the SILI bit is set, other methods for determining the actual block length should be used (e.g., including length information in the logical block).

If the SILI bit is one and the FIXED bit is one, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

If the SILI bit is zero and an incorrect-length logical block is read, CHECK CONDITION status shall be returned. The ILI and VALID bits shall be set to one in the sense data and the additional sense code shall be set to NO ADDITIONAL SENSE INFORMATION. Upon termination, the logical position shall be after the incorrect-length logical block (end-of-partition side). If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read (not including the incorrect-length logical block). If

the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length minus the actual logical block length. Logical units that do not support negative values shall set the INFORMATION field to zero if the overlength condition exists.

> NOTE 18 In the above case with the FIXED bit of one, only the position of the incorrect-length logical block may be determined from the sense data. The actual length of the incorrect logical block is not reported. Other means may be used to determine its actual length (e.g., read it again with the fixed bit set to zero).

A TRANSFER LENGTH of zero specifies no data shall be transferred, and the logical position shall not be changed. This condition shall not be considered an error.

In the case of an unrecovered read error, if the FIXED bit is one, the sense data VALID bit shall be set to one and the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read (not including the unrecovered logical blocks). If the FIXED bit is zero, the sense data VALID bit shall be set to one and the INFORMATION field shall be set to the requested transfer length. Upon termination, the logical position shall be after the unrecovered logical block.

If the device server encounters a filemark during a READ(6) command, CHECK CONDITION status shall be returned and the FILEMARK and VALID bits shall be set to one in the sense data. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate, and the additional sense code shall be set to FILEMARK DETECTED. Upon termination, the logical position shall be after the filemark (end-of-partition side). If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

If the device server encounters early-warning during a READ(6) command and the REW bit is set to one in the Device Configuration mode page (see 8.3.3), CHECK CONDITION status shall be returned upon completion of the current logical block. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate, and the additional sense code shall be set to END-OF-PARTITON/MEDIUM DETECTED. The EOM and VALID bits shall be set to one in the sense data. Upon termination, the logical position shall be after the last logical block transferred (end-of-partition side). If the FIXED bit is one, the INFORMATION field in the sense data shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field in the sense data shall be set to the requested transfer length minus the actual logical block length. The device server shall not return CHECK CONDITION status when early-warning is encountered if the REW bit is zero.

> NOTE 19 A REW bit of one is not recommended for most applications since read data may be present after early-warning.

If the device server encounters end-of-data during a READ(6) command, CHECK CONDITION status shall be returned, the sense key shall be set to BLANK CHECK, and the VALID bit shall be set to one in the sense data. If end-of-data is encountered at or after early-warning, the EOM bit shall also be set to one. Upon termination, the logical position shall be immediately after the last recorded logical object (end-of-partition side). If the FIXED bit is one, the INFORMATION field in the sense data shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

If the device server encounters end-of-partition during a READ(6) command, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, and the EOM and VALID bits shall be set to one in the sense data. The medium position following this condition is not defined. If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks read. If the FIXED bit is zero, the INFORMATION field in the sense data shall be set to the requested transfer length.

6.5 READ REVERSE(6) command

The READ REVERSE(6) command (see table 23) requests that the device server transfer one or more logical block(s) to the application client beginning at the current logical position. Prior to performing the read reverse operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 23 — READ REVERSE(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0Fh)							
1	Reserved					BYTORD	SILI	FIXED
2	(MSB)							
3				TRANSFER LENGTH				
4								(LSB)
5	CONTROL							

This command is similar to the READ(6) command except that medium motion is in the reverse direction. Upon completion of a READ REVERSE(6) command, the logical position shall be before the last logical block transferred (beginning-of-partition side).

A byte order (BYTORD) bit of zero specifies all logical block(s), and the byte(s) within the logical block(s), are transferred in the reverse order. The order of bits within each byte shall not be changed. A BYTORD bit of one specifies all logical block(s) are transferred in the reverse order but the byte(s) within the logical block(s) are transferred in the same order as returned by the READ(6) command. Support for either value of the BYTORD bit is optional.

Refer to the READ(6) command (see 6.4) for a description of the FIXED bit, the SILI bit, the TRANSFER LENGTH field, and any conditions that may result from incorrect usage of these fields.

Filemarks, incorrect-length logical blocks, and unrecovered read errors are handled the same as in the READ(6) command, except that upon termination the logical position shall be before the filemark, incorrect-length logical block, or unrecovered block (beginning-of-partition side).

If the device server encounters beginning-of-partition during a READ REVERSE(6) command, CHECK CONDITION status shall be returned and the EOM and VALID bits shall be set to one in the sense data. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks transferred. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

6.6 SPACE(6) command

The SPACE(6) command (see table 24) provides a variety of positioning functions that are determined by the CODE and COUNT fields. Both forward and reverse positioning are provided, although some logical units may only support a subset of this command. Prior to performing the space operation, except as stated in the description of the COUNT field, the device server shall perform a synchronize operation (see 4.2.8). If an application client requests an unsupported function, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB. The INFOR-MATION field value shall be equal to the magnitude of the COUNT field minus the magnitude of the logical objects

spaced over. A CHECK CONDITION caused by early termination of any SPACE(6) command shall not result in a negative INFORMATION field value.

Table 24 — SPACE(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (11h)							
1	Reserved				CODE			
2	(MSB)							
3				COUNT				
4								(LSB)
5	CONTROL							

The CODE field is defined in table 25.

Table 25 — Code definition

Code	Description	Support
0000b	Logical blocks	M
0001b	Filemarks	M
0010b	Sequential filemarks	O
0011b	End-of-data	O
0100b	Obsolete	
0101b	Obsolete	
0110b-1111b	Reserved	

When spacing over logical objects, the COUNT field specifies the number of logical objects to be spaced over in the current partition. A positive value N in the COUNT field when the CODE field is not 0011b (End-of-data) shall cause forward positioning (toward end-of-partition) over N logical objects ending on the end-of-partition side of the last logical object, if they exist. A zero value in the COUNT field when the CODE field is not 0011b (End-of-data) shall cause no change of logical position. A negative value $-N$ (two's complement notation) in the COUNT field when the CODE field is not 0011b (End-of-data) shall cause reverse positioning (toward beginning-of-partition) over N logical objects ending on the beginning-of-partition side of the last logical object, if they exist. When the CODE field is 0011b (End-of-data), the COUNT field shall be ignored and the device server shall perform a synchronize operation before moving before the end-of-data position. When the COUNT field is zero and the CODE field is not 0011b (End-of-data), a device server is not required to perform a synchronize operation. Support of spacing in the reverse direction is optional.

If a filemark is encountered while spacing over logical blocks, the command shall be terminated. CHECK CONDITION status shall be returned, and the FILEMARK and VALID bits shall be set to one in the sense data. The sense key shall be set to NO SENSE and the additional sense code shall be set to FILEMARK DETECTED. The INFORMATION field shall be set to the requested count minus the actual number of logical blocks spaced over. The logical position shall be on the end-of-partition side of the filemark if movement was in the forward direction and on the beginning-of-partition side of the filemark if movement was in the reverse direction.

If early-warning is encountered while spacing over logical objects and the REW bit is set to one in the Device Configuration mode page (see 8.3.3), CHECK CONDITION status shall be returned, the sense key shall be set to

NO SENSE, and the EOM and VALID bits shall be set to one in the sense data. The additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED. The INFORMATION field shall be set to the requested count minus the actual number of logical objects spaced over as defined by the CODE value. If the REW bit is zero, the device server shall not report CHECK CONDITION status at the early-warning point.

> NOTE 20 Setting the REW bit to one is not recommended for most applications since data may be present after early-warning.

If end-of-data is encountered while spacing over logical objects, CHECK CONDITION status shall be returned, the sense key shall be set to BLANK CHECK, and the sense data VALID bit shall be set to one in the sense data. The additional sense code shall be set to END-OF-DATA DETECTED. The sense data EOM bit shall be set to one if end-of-data is encountered at or after early-warning. The INFORMATION field shall be set to the requested count minus the actual number of logical objects spaced over as defined by the CODE value. The medium shall be positioned such that a subsequent write operation would append to the last logical object.

If the end-of-partition is encountered while spacing forward over logical objects, CHECK CONDITION status shall be returned, and the sense key shall be set to MEDIUM ERROR. The additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED, and the sense data EOM and VALID bit shall be set to one. The INFOR-MATION field shall be set to the requested count minus the actual number of logical objects spaced over as defined by the CODE value. The medium position following this condition is not defined.

If beginning-of-partition is encountered while spacing over logical objects in the reverse direction, the device server shall return CHECK CONDITION status and shall set the sense key to NO SENSE. The additional sense code shall be set to BEGINNING-OF-PARTITION/MEDIUM DETECTED. The sense data EOM and VALID bits shall be set to one, and the INFORMATION field set to the total number of logical objects not spaced over as defined by the CODE value (i.e., the requested number of logical objects minus the actual number of logical objects spaced over as defined by the CODE value). The medium position following this condition is not defined. A successfully completed SPACE(6) command shall not set EOM to one at beginning-of-partition.

When spacing over sequential filemarks, the count field is interpreted as follows:

  a) a positive value N shall cause forward movement to the first occurrence of N or more consecutive filemarks being logically positioned after the Nth filemark;
  b) a zero value shall cause no change in the logical position; or
  c) a negative value $-N$ (2's complement notation) shall cause reverse movement to the first occurrence of N or more consecutive filemarks being logically positioned on the beginning-of-partition side of the Nth filemark.

If end-of-partition is encountered while spacing to sequential filemarks, CHECK CONDITION status shall be returned, and the sense key shall be set to MEDIUM ERROR. The additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED, the EOM bit shall be set to one, and the VALID bit shall be set to zero in the sense data. The medium position following this condition is not defined.

If end-of-data is encountered while spacing to sequential filemarks, CHECK CONDITION status shall be returned, and the sense key shall be set to BLANK CHECK. The additional sense code shall be set to END-OF-DATA DETECTED, and the sense data VALID bit shall be set to zero. The medium shall be positioned such that a subse-quent write operation would append to the last logical object. The sense data EOM bit shall be set to one if end-of-data is encountered at or after early-warning.

When spacing to end-of-data, the COUNT field is ignored. Upon successful completion, the medium shall be positioned such that a subsequent write operation would append to the last logical object.

If end-of-partition is encountered while spacing to end-of-data, CHECK CONDITION status shall be returned, and the sense key shall be set to MEDIUM ERROR. The additional sense code shall be set to END-OF-PARTITION/

MEDIUM DETECTED, the EOM bit shall be set to one, and the VALID bit shall be set to zero in the sense data. The medium position following this condition is not defined.

6.7 VERIFY(6) command

The VERIFY(6) command (see table 26) requests that the device server verify one or more logical block(s) beginning at the current logical position. Prior to performing the verify operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 26 — VERIFY(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (13h)							
1	Reserved					IMMED	BYTCMP	FIXED
2	(MSB)							
3				VERIFICATION LENGTH				
4								(LSB)
5	CONTROL							

An immediate (IMMED) bit of zero specifies the command shall not return status until the verify operation has completed. An IMMED bit of one specifies status shall be returned as soon as the command descriptor block has been validated; but after all verification data has been transferred from the application client to the device server, if the BYTCMP bit is one.

> NOTE 21 In order to ensure that no errors are lost, the application client should set the IMMED bit to zero on the last VERIFY(6) command when issuing a series of VERIFY(6) commands.

A byte compare (BYTCMP) bit of zero specifies the verification shall be simply a medium verification (e.g., CRC, ECC). No data shall be transferred from the application client to the device server.

A BYTCMP bit of one specifies the device server shall perform a byte-by-byte compare of the data on the medium and the data transferred from the application client. Data shall be transferred from the application client to the device server as in a WRITE(6) command (see 6.8). If the BYTCMP bit is one and the byte compare option is not supported, the device server shall terminate the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

The VERIFICATION LENGTH field specifies the amount of data to verify, in logical blocks or bytes, as specified by the FIXED bit. Refer to the READ(6) command (see 6.4) for a description of the FIXED bit and any error conditions that may result from incorrect usage. If the VERIFICATION LENGTH field is zero, no data shall be verified and the current logical position shall not be changed. This condition shall not be considered as an error.

The VERIFY(6) command shall terminate as follows:

   a)   when the verification length has been satisfied;
   b)   when an incorrect-length logical block is encountered;
   c)   when a filemark is encountered;
   d)   when end-of-data is encountered;
   e)   when the end-of-partition is encountered;

   f)   when early-warning is encountered (if the REW bit is one in the Device Configuration mode page, see
        8.3.3); or
   g)   when an unrecoverable read error is encountered.

The status and sense data for each of the termination conditions are handled in the same manner as in the
READ(6) command (see 6.4). Upon successful completion of a VERIFY(6) command, the logical position shall be
after the last logical block verified.

If the data does not compare (BYTCMP bit of one), the command shall terminate with CHECK CONDITION status,
the sense data VALID bit shall be set to one, the sense key shall be set to MISCOMPARE, and the additional sense
code shall be set to MISCOMPARE DURING VERIFY OPERATION. If the FIXED bit is one, the INFORMATION field
shall be set to the requested verification length minus the actual number of logical blocks successfully verified. If
the FIXED bit is zero, the INFORMATION field shall be set to the requested verification length minus the actual number
of bytes successfully verified. This number may be larger than the requested verification length if the error occurred
on a previous VERIFY(6) command with an IMMED bit of one. Upon termination, the medium shall be positioned
after the logical block containing the miscompare (end-of-partition side).

6.8 WRITE(6) command

The WRITE command (see table 27) requests that the device server write the logical block that is transferred from
the application client to the current logical position.

Table 27 — WRITE(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0Ah)							
1	Reserved							FIXED
2	(MSB)							
3				TRANSFER LENGTH				
4								(LSB)
5	CONTROL							

The FIXED bit specifies whether fixed-block transfers or variable-block transfers are to be used. See the READ
BLOCK LIMITS command (see 7.4) for additional information about fixed-block transfers and variable-block
transfers.

If the FIXED bit is one, the TRANSFER LENGTH value specifies the number of fixed-length blocks to be transferred,
using the current block length reported in the mode parameter block descriptor (see 8.3). If the FIXED bit is zero, a
single logical block is transferred with TRANSFER LENGTH specifying the logical block length in bytes.

If TRANSFER LENGTH is zero, no data shall be transferred and the current position shall not be changed. This
condition shall not be considered an error.

A WRITE(6) command may be buffered or unbuffered, as specified by the BUFFERED MODE field of the mode
parameter header (see 8.3). When operating in unbuffered mode (see 3.1.70), the device server shall not return
GOOD status until all logical block(s) are successfully written to the medium. When operating in buffered mode
(see 3.1.8), the device server may return GOOD status as soon as all logical block(s) are successfully transferred
to the logical unit's object buffer.

NOTE 22 For compatibility with devices implemented prior to this version of this International Standard, a WRITE FILEMARKS command with the IMMED bit set to zero should be issued when completing a buffered write operation to perform a synchronize operation (see 4.2.8).

If the device server enables a WRITE(6) command while positioned between EW and EOP, or encounters EW during the processing of a WRITE(6) command, an attempt to finish writing any data may be made as determined by the current settings of the REW and SEW bits in the Device Configuration mode page (see 8.3.3). The command shall terminate with CHECK CONDITION status and the additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED. If all data that is to be written is successfully transferred to the medium, the sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If the device server is unable to transfer any data, buffered or unbuffered, when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW. If the SEW bit is set to zero, the EOM bit shall be set to one in the sense data. If the SEW bit is set to one, the EOM and VALID bits shall be set to one in the sense data.

The INFORMATION field shall be set as follows:

a) if the FIXED bit is set to one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks transferred to the device server; or
b) if the FIXED bit is set to zero, the INFORMATION field shall be set to the requested transfer length.

The device server should perform a synchronize operation (see 4.2.8) after the first early-warning indication has been returned to the application client (see 4.2.3).

NOTE 23 For some application clients it is important to recognize an error if end-of-partition is encountered during the processing of a WRITE(6) command, without regard for whether all data that is to be written is successfully transferred to the medium. The VOLUME OVERFLOW sense key may always validly be returned if end-of-partition is encountered while writing, and such usage is recommended. Reporting the MEDIUM ERROR sense key may cause confusion as to whether there was really defective medium encountered during the processing of the last WRITE(6) command.

If a WRITE(6) command is terminated early, an incomplete logical block (i.e., a logical block not completely transferred to the device server from the initiator) shall be discarded. The incomplete logical block may be accessible prior to new data being written to the media. The device server shall be logically positioned after the last logical block that was successfully transferred.

6.9 WRITE FILEMARKS(6) command

The WRITE FILEMARKS(6) command (see table 28) requests that the device server write the specified number of filemarks to the current position.

Table 28 — WRITE FILEMARKS(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (10h)							
1	Reserved						Obsolete	IMMED
2	(MSB)							
3	TRANSFER LENGTH							
4								(LSB)
5	CONTROL							

An immediate (IMMED) bit of one specifies the device server shall return status as soon as the command descriptor block has been validated. An IMMED bit of one is only valid if the device is operating in buffered mode (see 3.1.8). If the IMMED bit is set to one and the device is operating in unbuffered mode the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

An IMMED bit of zero specifies the device server shall not return status until the write operation has completed. Any buffered logical objects shall be written to the medium prior to completing the command.

> NOTE 24 Upon completion of any buffered write operation, the application client may issue a WRITE FILEMARKS(6) command with the IMMED bit set to zero and the TRANSFER LENGTH field set to zero to perform a synchronize operation (see 4.2.8).

If the device server enables a WRITE FILEMARKS(6) command while positioned between EW and EOP, or encounters EW during the processing of a WRITE FILEMARKS(6) command, an attempt to finish writing any buffered logical objects may be made, as determined by the current settings of the REW and SEW bits in the Device Configuration mode page (see 8.3.3). The command shall terminate with CHECK CONDITION status and the additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED. If all buffered logical objects to be written are successfully transferred to the medium, the sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If the device server is unable to transfer any buffered logical objects when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW. If the SEW bit is set to zero, the EOM bit shall be set to one in the sense data. If the SEW bit is set to one, the EOM and VALID bits shall be set to one in the sense data.

The INFORMATION field shall be set to the requested transfer length minus the actual number of filemarks:

a) written to the object buffer if the IMMED bit is set to one; or
b) written to the medium if the IMMED bit is set to zero.

The device server should perform a synchronize operation (see 4.2.8) after the first early-warning indication has been returned to the application client (see 4.2.3).

7 Common command descriptions for sequential-access devices

7.1 FORMAT MEDIUM command

The FORMAT MEDIUM command (see table 29) is used to prepare the medium for use by the logical unit. If buffered logical objects are stored by the device server when processing of a FORMAT MEDIUM command begins, the command shall be rejected with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to POSITION PAST BEGINNING OF MEDIUM.

Table 29 — FORMAT MEDIUM command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (04h)							
1	Reserved						VERIFY	IMMED
2	Reserved				FORMAT			
3	(MSB)							
4	TRANSFER LENGTH							(LSB)
5	CONTROL							

The FORMAT MEDIUM command shall be accepted only when the medium is at beginning-of-partition 0 (BOP 0). If the medium is logically at any other position, the command shall be rejected with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to POSITION PAST BEGINNING OF MEDIUM.

At the successful completion of a FORMAT MEDIUM command, the medium shall be positioned at BOP 0.

During the format operation, the device server shall respond to commands as follows:

  a) in response to all commands except REQUEST SENSE and INQUIRY, the device server shall return CHECK CONDITION unless a reservation conflict exists. In that case RESERVATION CONFLICT status shall be returned; or
  b) in response to the REQUEST SENSE command, assuming no error has occurred, the device server shall return a sense key of NOT READY and the additional sense code shall be set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS, with the sense key specific bytes set for process indication (see SPC-3).

An immediate (IMMED) bit of zero specifies the device server shall not return status until the FORMAT MEDIUM command has completed. An IMMED bit of one specifies the device server shall return status as soon as the valid medium location has been verified and the command descriptor block of the FORMAT MEDIUM command has been validated. If CHECK CONDITION status is returned for a FORMAT MEDIUM command with an IMMED bit of one, the format operation shall not be performed.

A VERIFY bit of one specifies the logical unit shall format the medium and then verify that the format was success-fully accomplished. The method used to verify success of the FORMAT MEDIUM command is vendor-specific. If the verify operation determines that the format was not successfully accomplished, the device server shall return a sense key of MEDIUM ERROR and the additional sense code shall be set to MEDIUM FORMAT CORRUPTED. If the VERIFY bit is zero, the logical unit shall not perform the verify check.

The FORMAT field is defined in table 30.

Table 30 — Format field definition

Value	Description	Support
0h	Use default format	O
1h	Partition medium	O
2h	Default format then partition	O
3h-7h	Reserved	
8h-Fh	Vendor-specific	

If the FORMAT field is 0h, the logical unit shall determine the format method to use. A valid FORMAT MEDIUM command with 0h in the FORMAT field shall cause all data on the entire physical volume to be lost.

If the FORMAT field is 1h, the logical unit shall partition the medium using the current mode data from the Medium Partition mode page (see 8.3.4). If none of the mode bits SDP, FDP, or IDP are set to one, the device server shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST with the addition sense code set to PARAMETER VALUE INVALID. If insufficient space exists on the medium for the requested partition sizes, the device server shall return CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR and the additional sense code shall be set to VOLUME OVERFLOW. A valid FORMAT MEDIUM command with 1h in the FORMAT field may cause all data on the entire physical volume to be lost.

If the FORMAT field is 2h, the logical unit shall perform the operations equivalent to a FORMAT field of 0h followed by a FORMAT field of 1h. A valid FORMAT MEDIUM command with 2h in the FORMAT field may cause all data on the entire physical volume to be lost.

When the FORMAT field contains 1h or 2h, some errors related to mode page field contents may not be detected until the FORMAT MEDIUM command is processed. Therefore, some error conditions described in 8.3.4 may be returned in response to a FORMAT MEDIUM command with 1h or 2h in the FORMAT field.

The TRANSFER LENGTH specifies the length in bytes of format information that shall transferred from the initiator. A transfer length of zero specifies no format information shall be transferred. This condition shall not be considered an error. If the FORMAT field is 0h, 1h, or 2h, the TRANSFER LENGTH shall be zero. Use of format information is restricted to vendor-specific values in the FORMAT field and the contents of the format information is vendor-specific.

7.2 LOAD UNLOAD command

The LOAD UNLOAD command (see table 31) requests that the logical unit enable or disable the logical unit for further operations. This command may also be used to request a retension function. Prior to performing the LOAD UNLOAD operation, the device server shall perform a synchronize operation (see 4.2.8). If the buffered mode is not 0h (see 8.3) and a previous command was terminated with CHECK CONDITION status and the device is

unable to continue successfully writing, the logical unit shall discard any unwritten buffered logical objects prior to performing the LOAD UNLOAD operation.

Table 31 — LOAD UNLOAD command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (1Bh)							
1	Reserved							IMMED
2	Reserved							
3	Reserved							
4	Reserved			HOLD	EOT	RETEN	LOAD	
5	CONTROL							

An immediate (IMMED) bit of zero specifies the device server shall not return status until the load or unload operation has completed. If the IMMED bit is one, the device server shall return status as soon as all buffered logical objects have been written to the medium and the command descriptor block of the LOAD UNLOAD command has been validated. If CHECK CONDITION status is returned for a LOAD UNLOAD command with an IMMED bit of one, the load or unload operation shall not be performed.

> NOTE 25 For compatibility with devices implemented prior to this version of the standard, a WRITE FILEMARKS command with an IMMED bit of zero should be used to perform a synchronize operation (see 4.2.8) prior to issuing a LOAD UNLOAD command with an IMMED bit of one.

A LOAD bit of one and a HOLD bit of zero specifies the medium in the logical unit shall be loaded and positioned to the beginning-of-partition zero. A LOAD bit of zero and a HOLD bit of zero specifies the medium in the logical unit shall be positioned for removal at the extreme position along the medium specified by the EOT bit. Following successful completion of an unload operation, the device server shall return CHECK CONDITION status with the sense key set to NOT READY for all subsequent medium-access commands until a new volume is mounted or a load operation is successfully completed.

A LOAD bit of one and a HOLD bit of one specifies if the medium has not been moved into the logical unit, the medium shall be moved in but not positioned for access. The EOT and RETEN bits shall be set to zero. Following successful completion, the device server shall return GOOD status. If both the medium and device server support MAM, the device server shall generate a unit attention condition for all initiators and the additional sense code shall be set to MEDIUM AUXILIARY MEMORY ACCESSIBLE.

A LOAD bit of zero and a HOLD bit of one specifies if the medium is in the logical unit, the medium shall be positioned as specified by the RETEN and EOT bits or shall be unthreaded (whichever is appropriate for the medium type) but shall not be de-mounted. Following successful completion, the device server shall return GOOD status.

A retension (RETEN) bit of one specifies the logical unit shall perform a retension function on the current medium. A RETEN bit of zero specifies the logical unit shall not perform a retension function on the current medium. Implementation of the retension function is vendor-specific.

An end-of-tape (EOT) bit of one specifies an unload operation (LOAD bit set to zero) shall position the medium at end-of-medium for removal from the device. For devices that do not support unloading at end-of-medium the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB. An EOT bit of zero specifies an unload operation shall position the medium at beginning-of-medium for removal from the device.

An EOT bit of one and a LOAD bit of one shall cause the device server to return CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

A HOLD bit of one specifies MAM shall be accessible upon completion of the command but the medium shall not be positioned for access. A HOLD bit of zero and a LOAD bit of one specifies the medium shall be positioned for access. A HOLD bit of zero and a LOAD bit of zero specifies MAM shall not be accessible upon completion of the command.

When operating in buffered mode 1h or 2h (see 8.3), the logical unit shall discard any unwritten buffered data after the LOAD UNLOAD command is validated if the device is unable to continue successfully writing (e.g., the device reported CHECK CONDITION status to a previous command, reported a write-type error, and the error has not been cleared or otherwise recovered).

7.3 LOCATE(16) command

The LOCATE(16) command (see table 32) causes the logical unit to position the medium to the logical object or logical file, as specified by the DEST_TYPE and LOGICAL IDENTIFIER fields. Upon completion, the logical position shall be as specified in table 33. Prior to performing the locate operation, the device server shall perform a synchronize operation (see 4.2.8).

Table 32 — LOCATE(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (92h)							
1	Reserved			DEST_TYPE		Rsvd	CP	IMMED
2	Reserved							BAM
3	PARTITION							
4	(MSB)							
5								
6								
7				LOGICAL IDENTIFIER				
8								
9								
10								
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	CONTROL							

An immediate (IMMED) bit of zero specifies the device server shall not return status until the locate operation has completed. If the IMMED bit is one, the device server shall return status as soon as all buffered logical objects have been written to the medium and the command descriptor block of the LOCATE(16) command has been validated. If

CHECK CONDITION status is returned for a LOCATE(16) command with an IMMED bit of one, the locate operation shall not be performed.

A change partition (CP) bit of one specifies a change to the partition specified in the PARTITION field shall occur prior to positioning to the logical object or logical file, as specified in the LOGICAL IDENTIFIER field. A CP bit of zero specifies no partition change shall occur and the PARTITION field shall be ignored.

The DEST_TYPE field shall be used in conjunction with the LOGICAL IDENTIFIER field to locate to the appropriate position of the medium. The DEST_TYPE field specifies whether the location specified is a logical object identifier or logical file identifier. The DEST_TYPE field is defined in table 33.

Table 33 — DEST_TYPE field definitions

Code	Description	Logical position upon successful completion	Support
00b	Logical object identifier	BOP side	M
01b	Logical file identifier	BOP side of the logical file	M
10b	Obsolete		
11b	Reserved		

A block address mode type (BAM) bit of zero specifies the logical unit shall process this command as an implicit address command. A BAM bit of one specifies the logical unit shall process this command as an explicit address command.

The LOGICAL IDENTIFIER field specifies the logical identifier to which the logical unit shall position the medium based on the current setting of the DEST_TYPE field. An otherwise valid LOCATE(16) command to any position between beginning-of-data and the position immediately after the last object in the partition (position at end-of-data) shall not return a sense key of ILLEGAL REQUEST. A LOCATE(16) to a position past end-of-data shall return CHECK CONDITION status and the sense key shall be set to BLANK CHECK. Additionally, the sense data EOM bit shall be set to one if end-of-data is located at or after early-warning.

If end-of-partition is encountered, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, and the sense data EOM bit shall be set to one.

The PARTITION field specifies the partition to select if the CP bit is one. Refer to the sequential-access device model (see 4.2.4) and the Medium Partition mode page (see 8.3.4) for additional information about partitioning.

The logical unit position is undefined if a LOCATE(16) command fails with a sense key other than ILLEGAL REQUEST.

7.4 READ BLOCK LIMITS command

The READ BLOCK LIMITS command (see table 34) requests that the READ BLOCK LIMITS data (see table 35) be returned. The READ BLOCK LIMITS data (see table 35) specifies the block length limits capability of the logical unit.

Table 34 — READ BLOCK LIMITS command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (05h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	CONTROL							

Table 35 — READ BLOCK LIMITS data

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved			GRANULARITY				
1	(MSB)							
2	MAXIMUM BLOCK LENGTH LIMIT							
3								(LSB)
4	(MSB)							
5	MINIMUM BLOCK LENGTH LIMIT							(LSB)

The GRANULARITY field specifies the supported block size granularity. The logical unit shall support all block sizes n such that n minus the minimum block length limit is a multiple of $2^{\text{GRANULARITY}}$ and n is greater than or equal to the MINIMUM BLOCK LENGTH LIMIT and less than or equal to the MAXIMUM BLOCK LENGTH LIMIT.

If the MAXIMUM BLOCK LENGTH LIMIT value equals the MINIMUM BLOCK LENGTH LIMIT value, the logical unit supports fixed-block transfers only, with the block length equal to the MINIMUM BLOCK LENGTH LIMIT value. In this case, READ and WRITE commands with the FIXED bit set to zero shall result in CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

For read and write commands with the FIXED bit set to one, block lengths are limited to multiples of four (see 8.3).

If the MAXIMUM BLOCK LENGTH LIMIT value is not equal to the MINIMUM BLOCK LENGTH LIMIT value, the logical unit supports fixed-block transfers or variable-block transfers, with the block length constrained between the given limits in either transfer mode. The transfer mode is controlled by the FIXED bit in the WRITE or READ commands. If the maximum block limit is zero, a maximum block length is not specified.

7.5 READ POSITION command

7.5.1 READ POSITION command description

The READ POSITION command (see table 36) reports the current position and provides information about logical objects contained in the object buffer. No medium movement shall occur as a result of responding to the command.

Table 36 — READ POSITION command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (34h)							
1	Reserved			SERVICE ACTION				
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)			ALLOCATION LENGTH				
8								(LSB)
9	CONTROL							

The service actions defined for the READ POSITION command are shown in table 37.

Table 37 — READ POSITION service action codes

Code	Name	Description	Implementation Requirements	Reference
00h	SHORT FORM -- BLOCK ID	Device server shall return 20 bytes of data with the FIRST LOGICAL OBJECT LOCATION and LAST LOGICAL OBJECT LOCATION fields as logical object identifier values (see 4.2.5.2), relative to a partition.	Mandatory	7.5.2
01h	SHORT FORM -- VENDOR-SPECIFIC	Device server shall return 20 bytes of data with the FIRST LOGICAL OBJECT LOCATION and LAST LOGICAL OBJECT LOCATION fields as vendor-specific values.	Optional	7.5.2
02h	Reserved	Illegal request		
03h	Reserved	Illegal request		
04h	Reserved	Illegal request		
05h	Reserved	Illegal request		
06h	LONG FORM	Device server shall return 32 bytes of data.	Mandatory	7.5.3
07h	Reserved	Illegal request		

Table 37 — READ POSITION service action codes (Continued)

Code	Name	Description	Implementation Requirements	Reference
08h	EXTENDED FORM	Device server shall return 28 bytes of data up to the maximum length specified by the ALLOCATION LENGTH field.	Optional	7.5.4
09h - 1Fh	Reserved	Illegal request		

If the device server does not implement the specified service action code, then the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

To maintain compatibility with earlier implementations, the service action codes 02h, 03h, 04h, 05h, and 07h shall not be implemented.

For service action codes of 00h, 01h, and 06h, the ALLOCATION LENGTH field in the CDB shall be zero. If it is not, then the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

For a service action code of 08h, the ALLOCATION LENGTH field in the CDB specifies how much space has been allocated for the parameter data. If the length is not sufficient to contain the parameter data, the first portion of the parameter data shall be returned. This shall not be considered an error. If the remainder of the parameter data is required, the application client should send a new READ POSITION command with an ALLOCATION LENGTH field large enough to contain all the parameter data.

7.5.2 READ POSITION data format, short form

Table 38 specifies the READ POSITION data that shall be returned if the SERVICE ACTION field is 00h or 01h.

Table 38 — READ POSITION data format, short form

Bit Byte	7	6	5	4	3	2	1	0
0	BOP	EOP	LOCU	BYCU	Rsvd	LOLU	PERR	Rsvd
1	PARTITION NUMBER							
2	Reserved							
3	Reserved							
4	(MSB)							
5				FIRST LOGICAL OBJECT LOCATION				
6								
7								(LSB)
8	(MSB)							
9				LAST LOGICAL OBJECT LOCATION				
10								
11								(LSB)
12	Reserved							
13	(MSB)							
14				NUMBER OF LOGICAL OBJECTS IN OBJECT BUFFER				
15								(LSB)
16	(MSB)							
17				NUMBER OF BYTES IN OBJECT BUFFER				
18								
19								(LSB)

A beginning–of-partition (BOP) bit of one specifies the logical unit is at the beginning-of-partition in the current partition. A BOP bit of zero specifies the current logical position is not at the beginning-of-partition.

An end-of-partition (EOP) bit of one specifies the logical unit is positioned between early-warning and end-of-partition in the current partition. An EOP bit of zero specifies the current logical position is not between early-warning and end-of-partition.

A logical object count unknown (LOCU) bit of one specifies the NUMBER OF LOGICAL OBJECTS IN OBJECT BUFFER field does not represent the actual number of logical objects in the object buffer. A LOCU bit of zero specifies the NUMBER OF LOGICAL OBJECTS IN OBJECT BUFFER field is valid.

A byte count unknown (BYCU) bit of one specifies the NUMBER OF BYTES IN OBJECT BUFFER field does not represent the actual number of bytes in the object buffer. A BYCU bit of zero specifies the NUMBER OF BYTES IN OBJECT BUFFER field is valid.

A logical object location unknown (LOLU) bit of one specifies the first logical object location, last logical object location, or partition number are not currently known or not otherwise obtainable. A LOLU bit of zero specifies the FIRST LOGICAL OBJECT LOCATION, LAST LOGICAL OBJECT LOCATION, and PARTITION NUMBER fields contain valid position information.

A position error (PERR) bit of one specifies the logical unit is unable to report the correct position due to an overflow of any of the returned position data. A PERR bit of zero specifies an overflow has not occurred in any of the returned position data fields.

The PARTITION NUMBER field reports the partition number for the current logical position. If the logical unit only supports one partition for the medium, the PARTITION NUMBER field shall be set to zero.

The FIRST LOGICAL OBJECT LOCATION field specifies the logical object identifier associated with the current logical position. The value shall specify the logical object identifier of the next logical object to be transferred between an application client and the device server if a READ or WRITE command is issued.

The LAST LOGICAL OBJECT LOCATION field specifies the logical object identifier associated with the next logical object to be transferred from the object buffer to the medium. If the object buffer does not contain a complete logical object or is empty, the value reported for the last logical object location shall be equal to the value reported for the first logical object location.

> NOTE 26 The information provided by the FIRST LOGICAL OBJECT LOCATION and LAST LOGICAL OBJECT LOCATION fields may be used in conjunction with the LOCATE command to position the medium at the appropriate logical object on another device in the case of unrecoverable errors on the first device.

The NUMBER OF LOGICAL OBJECTS IN OBJECT BUFFER field specifies the number of logical objects in the object buffer of the logical unit that have not been written to the medium.

The NUMBER OF BYTES IN OBJECT BUFFER field specifies the total number of data bytes in the object buffer of the logical unit that have not been written to the medium.

7.5.3 READ POSITION data format, long form

Table 39 specifies the format of the READ POSITION data that shall be returned if the SERVICE ACTION field is 06h.

Table 39 — READ POSITION data format, long form

Bit Byte	7	6	5	4	3	2	1	0
0	BOP	EOP	Reserved		MPU	LONU	Reserved	
1	Reserved							
2	Reserved							
3	Reserved							
4	(MSB)							
5				PARTITION NUMBER				
6								
7								(LSB)
8	(MSB)							
9								
10								
11				LOGICAL OBJECT NUMBER				
12								
13								
14								
15								(LSB)
16	(MSB)							
17								
18								
19				LOGICAL FILE IDENTIFIER				
20								
21								
22								
23								(LSB)
24	(MSB)							
25								
26								
27				Obsolete				
28								
29								
30								
31								(LSB)

The BOP, EOP, and PARTITION NUMBER fields are as defined in the READ POSITION data format, short form (see table 38).

A mark position unknown (MPU) bit of one specifies the logical file identifier is not known or accurate reporting is not currently available. A MPU bit of zero specifies the LOGICAL FILE IDENTIFIER field contains valid position information.

A logical object number unknown (LONU) bit of one specifies the logical object number or partition number are not known or accurate reporting is not currently available. A LONU bit of zero specifies the LOGICAL OBJECT NUMBER and PARTITION NUMBER fields contain valid position information.

The PARTITION NUMBER field reports the partition number for the current logical position. If the logical unit only supports one partition for the medium, the PARTITION NUMBER field shall be set to zero.

The LOGICAL OBJECT NUMBER field specfies the number of logical objects between beginning-of-partition and the current logical position. A filemark counts as one logical object.

The LOGICAL FILE IDENTIFIER field specifies the number of filemarks between beginning-of-partition and the current logical position. This value is the current logical file identifier.

7.5.4 READ POSITION data format, extended form

Table 40 specifies the format of the READ POSITION data that shall be returned if the SERVICE ACTION field is 08h.

Table 40 — READ POSITION data format, extended form

Bit Byte	7	6	5	4	3	2	1	0
0	BOP	EOP	LOCU	BYCU	Rsvd	LOPU	PERR	Rsvd
1	PARTITION NUMBER							
2	(MSB)	ADDITIONAL LENGTH (1Ch)						
3								(LSB)
4	Reserved							
5	(MSB)	NUMBER OF LOGICAL OBJECTS IN OBJECT BUFFER						
6								
7								(LSB)
8	(MSB)	FIRST LOGICAL OBJECT LOCATION						
9								
10								
11								
12								
13								
14								
15								(LSB)
16	(MSB)	LAST LOGICAL OBJECT LOCATION						
17								
18								
19								
20								
21								
22								
23								(LSB)
24	(MSB)	NUMBER OF BYTES IN OBJECT BUFFER						
25								
26								
27								
28								
29								
30								
31								(LSB)

The fields are defined the same as for the corresponding fields in the READ POSITION data format, short form (see table 38).

The ADDITIONAL LENGTH field shall contain 1Ch. If the information transferred to the Data-In Buffer is truncated because of an insufficient ALLOCATION LENGTH value, the ADDITIONAL LENGTH field shall not be altered to reflect the truncation.

7.6 RECOVER BUFFERED DATA command

The RECOVER BUFFERED DATA command (see table 41) is used to recover data that has been transferred to the logical unit's object buffer but has not been successfully written to the medium. It is normally used to recover the buffered data after error or exception conditions make it impossible to write the buffered data to the medium. One or more RECOVER BUFFERED DATA commands may be required to recover all unwritten buffered data.

Table 41 — RECOVER BUFFERED DATA command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (14h)							
1	Reserved						SILI	FIXED
2	(MSB)							
3				TRANSFER LENGTH				
4								(LSB)
5	CONTROL							

The processing of this command is similar to the READ(6) command except that the data is transferred from the logical unit's object buffer instead of the medium. The order that logical block(s) are transferred is defined by the RBO bit in the Device Configuration mode page (see 8.3.3). If the RBO bit is not implemented, logical block(s) are transferred in the same order they would have been transferred to the medium.

Refer to the READ(6) command (see 6.4) for a description of the FIXED bit, the SILI bit, the TRANSFER LENGTH field, and any conditions that may result from incorrect usage of these fields.

If the FIXED bit is zero, no more than the requested transfer length shall be transferred to the application client. If the requested transfer length is smaller than the actual length of the logical block to be recovered, only the requested transfer length shall be transferred to the application client and the remaining data for the current logical block shall be discarded.

> NOTE 27 During recovery operations involving unknown block sizes, the application client should select the maximum block length supported by the logical unit to ensure that all buffered data will be transferred and set the FIXED bit to zero.

If a buffered filemark is encountered during a RECOVER BUFFERED DATA command, CHECK CONDITION status shall be returned, the sense key shall be set to NO SENSE, and the FILEMARK and VALID bits shall be set to one in the sense data. Upon termination, the logical position shall be after the filemark. If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks transferred. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

If an attempt is made to recover more logical blocks of data than are contained in the logical unit's object buffer, CHECK CONDITION status shall be returned, the sense key shall be set to NO SENSE. The additional sense code

shall be set to END-OF-DATA DETECTED, and the EOM and VALID bits shall be set to one in the sense data. If the FIXED bit is one, the INFORMATION field shall be set to the requested transfer length minus the actual number of logical blocks transferred. If the FIXED bit is zero, the INFORMATION field shall be set to the requested transfer length.

7.7 REPORT DENSITY SUPPORT command

7.7.1 REPORT DENSITY SUPPORT command description

The REPORT DENSITY SUPPORT command (see table 42) requests that information regarding the supported densities or medium type for the logical unit be sent to the application client.

Table 42 — REPORT DENSITY SUPPORT command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (44h)							
1	Reserved						MEDIUM TYPE	MEDIA
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)			ALLOCATION LENGTH				
8								(LSB)
9	CONTROL							

If the MEDIA bit is set to zero, the device server shall return descriptors for densities or medium types supported by the logical unit for any supported media. If the MEDIA bit is set to one, the device server shall return descriptors for densities or the medium type supported by the mounted medium. If the MEDIA bit is set to one and the logical unit either contains no medium or contains a medium but cannot determine the medium's density or type, CHECK CONDITION status shall be returned. The sense key shall be set to NOT READY and the additional sense code shall specify the reason for NOT READY.

If the MEDIUM TYPE bit is set to zero, the device server shall return data as specified in 7.7.3. If the MEDIUM TYPE bit is set to one, the device server shall return data as specified in 7.7.4.

The ALLOCATION LENGTH field specifies the maximum number of bytes that the device server may return.

7.7.2 REPORT DENSITY SUPPORT header

The REPORT DENSITY SUPPORT header is specified in table 43.

Table 43 — REPORT DENSITY SUPPORT header

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)			AVAILABLE DENSITY SUPPORT LENGTH				
1								(LSB)
2	Reserved							
3	Reserved							
4	Descriptor(s)							
n								

The AVAILABLE DENSITY SUPPORT LENGTH field specifies the number of bytes in the following data that is available to be transferred. The available density support length does not include itself. If the parameter data is truncated due to insufficient allocation length, the AVAILABLE DENSITY SUPPORT LENGTH field shall not be altered to reflect the truncation.

7.7.3 Density support report

The REPORT DENSITY SUPPORT command with a MEDIUM TYPE bit set to zero returns the REPORT DENSITY SUPPORT header (see table 43) followed by one or more density support data block descriptors (see table 44). The density support data block descriptors shall follow the REPORT DENSITY SUPPORT header. The density support data block descriptors shall be in numerical ascending order of the PRIMARY DENSITY CODE value.

Table 44 — Density support data block descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	PRIMARY DENSITY CODE							
1	SECONDARY DENSITY CODE							
2	WRTOK	DUP	DEFLT	Reserved				DLV
3	(MSB)	DESCRIPTOR LENGTH						
4								(LSB)
5	(MSB)							
6		BITS PER MM						
7								(LSB)
8	(MSB)	MEDIA WIDTH						
9								(LSB)
10	(MSB)	TRACKS						
11								(LSB)
12	(MSB)							
13		CAPACITY						
14								
15								(LSB)
16 : 23	ASSIGNING ORGANIZATION							
24 : 31	DENSITY NAME							
32 : 51	DESCRIPTION							

Density support data block descriptors shall be returned by ascending PRIMARY DENSITY CODE values. Multiple entries may exist for a given PRIMARY DENSITY CODE value. For all entries with equal PRIMARY DENSITY CODE values, all fields except for ASSIGNING ORGANIZATION, DENSITY NAME, and DESCRIPTION shall be identical. Density support data block descriptors with the same PRIMARY DENSITY CODE value should be ordered from most to least preferred ASSIGNING ORGANIZATION, DENSITY NAME, and DESCRIPTION.

> NOTE 28 By allowing multiple entries for a given primary and secondary density code set, multiple standard names may identify the same density code. This facilitates the remapping of density codes, if required.

The density support data block descriptor may represent a particular format in addition to giving physical density information. The information in a density support data block descriptor provides an application client with a detailed review of the recording technologies supported by a logical unit. By supplying the density code value returned in a

density support data block descriptor in a MODE SELECT command (see 8.3), an application client selects the recording technology (density, format, etc.).

The PRIMARY DENSITY CODE field contains the value returned by a MODE SENSE command for the density described in the remainder of the density support data block descriptor. The device server shall accept a MODE SELECT command containing this value, for appropriate media. The value 7Fh shall be reserved. All other values are available for use. The value of 00h shall only be used for the default density of the logical unit.

When multiple density code values are assigned to the same recording technology (density, format, etc.), the SECONDARY DENSITY CODE field shall contain the equivalent density code value. When the SECONDARY DENSITY CODE is used in the mode select header with a MODE SELECT command, the device server shall accept this value as equivalent to the PRIMARY DENSITY CODE value. If no secondary density code exists, the device server shall return the PRIMARY DENSITY CODE value in this field.

A WRTOK bit of zero specifies the logical unit does not support writing to the media with this density. A WRTOK bit of one specifies the logical unit is capable of writing this density to either the currently mounted medium (MEDIA bit in command descriptor block set to one) or for some media (MEDIA bit in command descriptor block set to zero). All density code values returned by the REPORT DENSITY SUPPORT command shall be supported for read operations.

A DUP bit of zero specifies this primary density code has exactly one density support data block descriptor. A DUP bit of one specifies this primary density code is specified in more than one density support data block descriptor.

A DEFLT bit of zero specifies this density is not the default density of the drive. A DEFLT bit of one specifies this density is the default density. If either the PRIMARY DENSITY CODE or the SECONDARY DENSITY CODE field is zero, the DEFLT bit shall be one. If neither the primary or secondary density code is zero and the DEFLT bit is one, the logical unit shall accept a MODE SELECT header with a density code of 00h as equivalent to the primary and secondary density codes.

> NOTE 29 The default density of the logical unit may vary depending on the currently mounted media. Multiple codes may return a DEFLT bit of one when the MEDIA bit is zero since more than one default may be possible.

If the Descriptor Length Valid (DLV) bit is set to one, the DESCRIPTOR LENGTH field shall contain the length of the descriptor minus 5. If the DLV bit is set to zero, the DESCRIPTOR LENGTH field shall be set to zero and the descriptor is 52 bytes in length.

The BITS PER MM field specifies the number of bits per millimeter per track as recorded on the medium. The value in this field shall be rounded up if the fractional value of the actual value is greater than or equal to 0,5. A value of 00h specifies the number of bits per millimeter does not apply to this logical unit. Direct comparison of this value between different vendors (possibly products) is discouraged since the definition of bits may vary.

The MEDIA WIDTH field specifies the width of the medium supported by this density. This field has units of tenths of millimeters. The value in this field shall be rounded up if the fractional value of the actual value is greater than or equal to 0,5. The MEDIA WIDTH field may vary for a given density depending on the mounted medium. A value of 00h specifies the width of the medium does not apply to this logical unit.

The TRACKS field specifies the number of data tracks supported on the medium by this density. The TRACKS value may vary for a given density depending on the mounted medium. Direct comparison of this value between different vendors (possibly products) is discouraged since the definition of the number of tracks may vary. For recording formats that are neither parallel nor serpentine, the TRACKS field specifies the maximum number of data tracks that are read or recorded simultaneously.

If the MEDIA bit is zero, the CAPACITY field specifies the approximate capacity of the longest supported medium assuming recording in this density with one partition. If the MEDIA bit is one, the CAPACITY field should specify the

approximate capacity of the current medium, assuming recording in this density with one partition. If the approximate capacity of the current medium is not available for the mounted medium, the longest supported medium capacity shall be used. If a SET CAPACITY command has affected the capacity of the medium, this shall be reflected in the CAPACITY field. The capacity assumes that compression is disabled, if possible. If this density does not support an uncompressed format, the capacity assumes that compression is enabled using average data. The capacity also assumes that the media is in good condition, and that normal data and block sizes are used. This value is in units of megabytes (10^6 bytes). The logical unit does not guarantee that this space is actually available in all cases. Direct comparison of this value between different vendors (possibly products) is discouraged since the length of media and the method used to measure maximum capacity may vary. The CAPACITY field is intended to be used by the application client to determine that the correct density is being used, particularly when a lower-density format is required for interchange.

The ASSIGNING ORGANIZATION field contains eight bytes of ASCII data identifying the organization responsible for the specifications defining the values in this density support data block descriptor. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required. The ASSIGNING ORGANIZATION field should contain a value listed in the vendor identification list (see SPC-3). The use of a specific vendor identification, other than the one associated with the device is allowed.

> NOTE 30 If vendor X defines a density and format, another vendor may use X in the ASSIGNING ORGANIZATION field. If exactly the same density and format construction later becomes known by another name, both X and the new assigning organization may be used for the density code. This is one condition that may result in multiple density support data block descriptors for a single density code value.

> NOTE 31 It is intended that the ASSIGNING ORGANIZATION field contain a unique identification of the organization responsible for the information in a density support data block descriptor. In the absence of any formal registration procedure, T10 maintains a list of vendor and assigning organization identification codes in use. Vendors are requested to voluntarily submit their identification codes to prevent duplication of codes.

The DENSITY NAME field contains eight bytes of ASCII data identifying the document (or other identifying name) that is associated with this density support data block descriptor. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required. Two physical densities (and possibly formats) shall not have identical ASSIGNING ORGANIZATION and DENSITY NAME fields. Assigning organizations are responsible for preventing duplicate usage of one density name for multiple different densities and/or formats.

> NOTE 32 It is suggested that any document that specifies a format and density for the media contain the values to be used by a logical unit when reporting the density support. The values for the BITS PER MM, MEDIA WIDTH, and TRACKS should also be included in such a document to help maintain consistency.

The DESCRIPTION field contains twenty bytes of ASCII data describing the density. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required.

7.7.4 Medium Type support report

The REPORT DENSITY SUPPORT command with a MEDIUM TYPE field bit set to one returns the REPORT DENSITY SUPPORT header (see table 43) followed by one or more medium type descriptors (see table 45). The

medium type descriptors shall follow the REPORT DENSITY SUPPORT header. The medium type descriptors shall be in numerical ascending order of the medium type value.

Table 45 — Medium type descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	MEDIUM TYPE							
1	Reserved							
2	(MSB)			DESCRIPTOR LENGTH (52)				
3								(LSB)
4	NUMBER OF DENSITY CODES							
5	(MSB)							(LSB)
:	(MSB)			PRIMARY DENSITY CODES				(LSB)
13	(MSB)							(LSB)
14	(MSB)			MEDIA WIDTH				
15								(LSB)
16	(MSB)			MEDIUM LENGTH				
17								(LSB)
18	Reserved							
19	Reserved							
20	ASSIGNING ORGANIZATION							
:								
27								
28	MEDIUM TYPE NAME							
:								
35								
36	DESCRIPTION							
:								
55								

The MEDIUM TYPE field contains the value returned by a MODE SENSE command for the medium type described in the remainder of the medium type descriptor. The device server shall accept a MODE SELECT command containing this value, for appropriate media.

The DESCRIPTOR LENGTH field contains the length of the descriptor minus 4.

The NUMBER OF DENSITY CODES field contains the number of valid density codes present in the PRIMARY DENSITY CODES value field.

The PRIMARY DENSITY CODES field contains a list of primary density code values supported by the drive for the medium type. The primary density code values shall be listed in ascending order. Any unused bytes in this field shall contain zero.

The MEDIA WIDTH field specifies the width of the medium. This field has units of tenths of millimeters. The value in this field shall be rounded up if the fractional portion of the actual value is greater than or equal to 0,5.

The MEDIUM LENGTH field specifies the nominal length of the medium. This field has units of meters. The value in this field shall be rounded up if the fractional portion of the actual value is greater than or equal to 0,5.

The ASSIGNING ORGANIZATION field contains eight bytes of ASCII data identifying the organization responsible for the specifications defining the values in this medium type descriptor. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required. The ASSIGNING ORGANIZATION field should contain a value listed in the vendor identification list (see SPC-3). The use of a vendor identification other than the one associated with the device is allowed.

> NOTE 33 It is intended that the ASSIGNING ORGANIZATION field contain a unique identification of the organization responsible for the information in a medium type descriptor. In the absence of any formal registration procedure, T10 maintains a list of vendor and assigning organization identification codes in use. Vendors are requested to voluntarily submit their identification codes to prevent duplication of codes.

The MEDIUM TYPE NAME field contains eight bytes of ASCII data identifying the document (or other identifying name) that is associated with this medium type descriptor. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required. Two different medium types shall not have identical ASSIGNING ORGANIZATION and MEDIUM TYPE NAME fields. Assigning organizations are responsible for preventing duplicate usage of one medium type name for multiple different medium types.

> NOTE 34 It is suggested that any document that specifies a characteristics for the media contain the values to be used by a logical unit when reporting the density support. The values for the MEDIUM WIDTH and MEDIUM LENGTH should also be included in such a document to help maintain consistency.

The DESCRIPTION field contains twenty bytes of ASCII data describing the medium type. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required.

7.8 REWIND command

The REWIND command (see table 46) causes the logical unit to position to the beginning-of-partition in the current partition. Prior to performing the rewind operation, the device server shall perform a synchronize operation (see 4.2.8). If the buffered mode is not 0h (see 8.3) and a previous command was terminated with CHECK CONDITION status and the device is unable to continue successfully writing, the logical unit shall discard any unwritten buffered logical objects prior to performing the REWIND operation.

Table 46 — REWIND command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (01h)							
1	Reserved							IMMED
2	Reserved							
3	Reserved							
4	Reserved							
5	CONTROL							

An immediate (IMMED) bit of zero specifies the device server shall not return status until the rewind operation has completed. If the IMMED bit is one, the device server shall return status as soon as all buffered logical objects have been written to the medium and the command descriptor block of the REWIND command has been validated. If CHECK CONDITION status is returned for a REWIND command with an IMMED bit of one, the rewind operation shall not be performed.

> NOTE 35 For compatibility with devices implemented prior to this standard, it is suggested that a WRITE FILEMARKS command with an IMMED bit of zero be used to perform a synchronize operation (see 4.2.8) before issuing a REWIND command with an IMMED bit of one.

7.9 SET CAPACITY command

The SET CAPACITY command (see table 47) sets the available medium for the currently mounted volume to a proportion of the total capacity of that volume. Any excess space shall be unavailable on the volume after successful completion of this command until changed by a new SET CAPACITY command. This change shall persist through power cycles, logical unit resets, I_T nexus losses, and unloading or reloading of the volume. Other vendor-specific actions such as physical erasure may change the total capacity of the volume. The method for recording the available capacity and other marks needed to manage the resulting capacity for volume interchange may be specified in a recording format standard or may be vendor-specific.

Table 47 — SET CAPACITY command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0Bh)							
1	Reserved							IMMED
2	Reserved							
3	(MSB)			CAPACITY PROPORTION VALUE				
4								(LSB)
5	CONTROL							

If the device server does not contain a medium, then the command shall be terminated with CHECK CONDITION status. The sense key shall be set to NOT READY, and the additional sense code shall be set to MEDIUM NOT PRESENT.

The SET CAPACITY command shall be accepted only when the medium is at beginning-of-partition 0 (BOP 0). If the medium is logically at any other position, the command shall be rejected with CHECK CONDITION status. The sense key shall be ILLEGAL REQUEST and the additional sense code shall be set to POSITION PAST BEGINNING OF MEDIUM.

A valid SET CAPACITY command shall cause all data and partitioning information on the entire physical volume to be lost. If the partitioning information changes, the device server shall generate a unit attention condition for all initiators with the additional sense code set to MODE PARAMETERS CHANGED.

An immediate (IMMED) bit of zero specifies the device server shall not return status until the set capacity operation has completed. An IMMED bit of one specifies the device server shall return status as soon as the command descriptor block of the SET CAPACITY command has been validated. If CHECK CONDITION status is returned for a SET CAPACITY command with an IMMED bit set to one, the set capacity operation shall not be performed.

The CAPACITY PROPORTION VALUE field specifies the portion of the total volume capacity to be made available for use. The CAPACITY PROPORTION VALUE field is the numerator to a fraction with a denominator of 65 535. The resulting available capacity on the volume shall be equal to the total volume capacity multiplied by this fraction. The device server may round up the capacity to the next highest supported value. This rounding shall not be considered an error and shall not be reported.

> NOTE 36 Available and total volume capacities are approximate values that may be affected by defects that reduce the actual available capacity of the volume. Other factors, such as partitioning, compression, and logical block packing may also affect available capacity.

7.10 SPACE(16) command

The SPACE(16) command (see table 48) operates identically to the SPACE(6) command (see 6.6), but allows specifying a COUNT field up to eight bytes in length and has parameter data out that specifies the logical object identifier on the medium. Following completion of a SPACE(16) command a READ POSITION command should be issued to obtain positioning information.

Table 48 — SPACE(16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (91h)							
1	Reserved				CODE			
2	Reserved							
3	Reserved							
4	(MSB)							
5								
6								
7				COUNT				
8								
9								
10								
11								(LSB)
12	(MSB)			PARAMETER LENGTH				
13								(LSB)
14	Reserved							
15	CONTROL							

The CODE field is defined in table 25 (see 6.6).

When spacing over logical objects, the COUNT field specifies the number of logical objects to be spaced over in the current partition. A positive value N in the COUNT field when the CODE field is not 0011b (End-of-data) shall cause forward positioning (toward end-of-partition) over N logical objects ending on the end-of-partition side of the last logical object, if they exist. A zero value in the COUNT field when the CODE field is not 0011b (End-of-data) shall cause no change of logical position. A negative value $-N$ (two's complement notation) in the COUNT field when the CODE field is not 0011b (End-of-data) shall cause reverse positioning (toward beginning-of-partition) over N logical

objects ending on the beginning-of-partition side of the last logical object, if they exist. When the CODE field is 0011b (End-of-data), the COUNT field shall be ignored and the device server shall perform a synchronize operation before moving before the end-of-data position. When the COUNT field is zero and the CODE field is not 0011b (End-of-data), a device server is not required to perform a synchronize operation. Support of spacing in the reverse direction is optional.

The PARAMETER LENGTH field is used to send parameter data space positioning information specifying the position on the medium from which to start the SPACE(16) command function. For an implicit block address mode command, the PARAMETER LENGTH field shall be set to 0. For an explicit block address mode command, the PARAMETER LENGTH field shall be set to 16. If the PARAMETER LENGTH field is set to any other value, the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

Space positioning information is specified in table 49.

Table 49 — Space positioning information

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	Reserved							
3	PARTITION NUMBER							
4	(MSB)							
5								
6								
7				LOGICAL OBJECT IDENTIFIER				
8								
9								
10								
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	Reserved							

The LOGICAL OBJECT IDENTIFIER and PARTITION fields specify the position where the SPACE(16) command shall start. If the current logical position does not match the specified LOGICAL OBJECT IDENTIFIER and PARTITION fields, the device server shall perform a locate operation to the specified logical object identifier and partition prior to performing the space operation. If the locate operation fails, the device server shall return CHECK CONDITION status and the additional sense code shall be set to LOCATE OPERATION FAILURE. Following a locate operation failure the logical position is undefined.

> NOTE 37 Locating to the logical object identifier prior to performing the space operation is necessary for the space operation to function properly when filemarks are between the starting logical object identifier and the expected ending logical object identifier of the space operation.

If a filemark is encountered while spacing over logical blocks, the command shall be terminated. CHECK CONDITION status shall be returned, and the FILEMARK and VALID bits shall be set to one in the sense data. The sense key shall be set to NO SENSE and the additional sense code shall be set to FILEMARK DETECTED. The INFORMATION field shall be set to the requested count minus the actual number of logical blocks spaced over. The logical position shall be on the end-of-partition side of the filemark if movement was in the forward direction and on the beginning-of-partition side of the filemark if movement was in the reverse direction.

> NOTE 38 For some space operations using the SPACE(16) command, the INFORMATION field value may exceed the maximum value allowed in the fixed format sense data (see SPC-3). As such, the descriptor format sense data (see SPC-3) should be enabled (i.e., the D_SENSE bit is set to one in the Control mode page).

If early-warning is encountered while spacing over logical objects and the REW bit is set to one in the Device Configuration mode page (see 8.3.3), CHECK CONDITION status shall be returned, the sense key shall be set to NO SENSE, and the EOM and VALID bits shall be set to one in the sense data. The additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED. The INFORMATION field shall be set to the requested count minus the actual number of logical objects spaced over as defined by the CODE value. If the REW bit is zero or the option is not supported by the logical unit, the device server shall not report CHECK CONDITION status at the early-warning point.

> NOTE 39 Setting the REW bit to one is not recommended for most applications since data may be present after early-warning.

If end-of-data is encountered while spacing over logical objects, CHECK CONDITION status shall be returned, the sense key shall be set to BLANK CHECK, and the sense data VALID bit shall be set to one in the sense data. The additional sense code shall be set to END-OF-DATA DETECTED. The sense data EOM bit shall be set to one if end-of-data is encountered at or after early-warning. The INFORMATION field shall be set to the requested count minus the actual number of logical objects spaced over as defined by the CODE value. The medium shall be positioned such that a subsequent write operation would append to the last logical object.

If the end-of-partition is encountered while spacing forward over logical objects, CHECK CONDITION status shall be returned, and the sense key shall be set to MEDIUM ERROR. The additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED, and the sense data EOM and VALID bit shall be set to one. The INFOR-MATION field shall be set to the requested count minus the actual number of logical objects spaced over as defined by the CODE value. The medium position following this condition is not defined.

If beginning-of-partition is encountered while spacing over logical objects in the reverse direction, the device server shall return CHECK CONDITION status and shall set the sense key to NO SENSE. The additional sense code shall be set to BEGINNING-OF-PARTITION/MEDIUM DETECTED. The sense data EOM and VALID bits shall be set to one, and the INFORMATION field set to the total number of logical objects not spaced over as defined by the CODE value (i.e., the requested number of logical objects minus the actual number of logical objects spaced over as defined by the CODE value). The medium position following this condition is not defined. A successfully completed SPACE(16) command shall not set EOM to one at beginning-of-partition.

When spacing over sequential filemarks, the count field is interpreted as follows:

   a) a positive value N shall cause forward movement to the first occurrence of N or more consecutive filemarks being logically positioned after the Nth filemark;
   b) a zero value shall cause no change in the logical position; or
   c) a negative value $-N$ (2's complement notation) shall cause reverse movement to the first occurrence of N or more consecutive filemarks being logically positioned on the beginning-of-partition side of the Nth filemark.

If end-of-partition is encountered while spacing to sequential filemarks, CHECK CONDITION status shall be returned, and the sense key shall be set to MEDIUM ERROR. The additional sense code shall be set to

END-OF-PARTITION/MEDIUM DETECTED, the EOM bit shall be set to one, and the VALID bit shall be set to zero in the sense data. The medium position following this condition is not defined.

If end-of-data is encountered while spacing to sequential filemarks, CHECK CONDITION status shall be returned, and the sense key shall be set to BLANK CHECK. The additional sense code shall be set to END-OF-DATA DETECTED, and the sense data VALID bit shall be set to zero. The medium shall be positioned such that a subsequent write operation would append to the last logical object. The sense data EOM bit shall be set to one if end-of-data is encountered at or after early-warning.

When spacing to end-of-data, the COUNT field is ignored. Upon successful completion, the medium shall be positioned such that a subsequent write operation would append to the last logical object.

If end-of-partition is encountered while spacing to end-of-data, CHECK CONDITION status shall be returned, and the sense key shall be set to MEDIUM ERROR. The additional sense code shall be set to END-OF-PARTITION/ MEDIUM DETECTED, the EOM bit shall be set to one, and the VALID bit shall be set to zero in the sense data. The medium position following this condition is not defined.

8 Parameters for sequential-access devices

8.1 Diagnostic parameters

This subclause defines the descriptors and pages for diagnostic parameters used with sequential-access devices.

The diagnostic page codes for sequential-access devices are defined in table 50.

Table 50 — Diagnostic page codes

Page Code	Description	Reference
00h	Supported diagnostic pages	SPC-3
01h - 3Fh	Reserved (for all device types)	
40h - 7Fh	Reserved	
80h - FFh	Vendor-specific	

8.2 Log parameters

8.2.1 Log parameters overview

This subclause defines the descriptors and pages for log parameters used with sequential-access devices.

The log page codes for sequential-access devices are defined in table 51.

Table 51 — Log page codes

Page Code	Description	Reference
00h	Supported log pages	SPC-3
01h	Buffer Overrun/Underrun log page	SPC-3
02h	Write Error Counter log page	SPC-3
03h	Read Error Counter log page (read)	SPC-3
04h	Read Reverse Error Counter log page	SPC-3
05h	Verify Error Counter log page	SPC-3
06h	Non-Medium Error log page	SPC-3
07h	Last n Error Events log page	SPC-3
08h	Format Status log page	SPC-3
09h - 0Ah	Reserved	
0Bh	Last n Deferred Error Events log page	SPC-3
0Ch	Sequential-Access Device log page	8.2.2
0Dh	Temperature log page	SPC-3
0Eh	Start-Stop Cycle Counter log page	SPC-3
a) Optional for SSC-3 devices, mandatory for ADC devices.		

Table 51 — Log page codes (Continued)

Page Code	Description	Reference
0Fh	Application Client log page	SPC-3
10h	Self-test Results log page	SPC-3
11h	DTD Status log page	ADC[a]
11h - 2Dh	Reserved	
2Eh	TapeAlert log page	8.2.3
2Fh	Informational Exceptions log page	SPC-3
30h - 3Fh	Vendor-specific (does not require page format)	
a) Optional for SSC-3 devices, mandatory for ADC devices.		

8.2.2 Sequential-Access Device log page

The Sequential-Access Device log page defines data counters associated with data bytes transferred to and from the media and to and from the application client, and a list parameter of binary information on cleaning.

The default value for parameters 0 through 3 shall be zero.

> NOTE 40 The data in parameters 0 and 1 are intended to provide an indication of the compression ratio for the written data. Parameters 2 and 3 are intended to provide an indication of the compression ratio for read data.

Support of the Sequential-Access Device log page is optional. Support of the individual parameters in the Sequential-Access Device log page are optional.

Table 52 defines the parameter codes for the Sequential-Access Device log page.

Table 52 — Parameter codes for Sequential-Access Device log page

Parameter Code	Description
0000h	Number of data bytes received from application clients during WRITE command operations.
0001h	Number of data bytes written to the media as a result of WRITE command operations, not counting ECC and formatting overhead.
0002h	Number of data bytes read from the media during READ command operations, not counting ECC and formatting overhead.
0003h	Number of data bytes transferred to the initiator(s) during READ command operations.
0004h - 00FFh	Reserved.
0100h	Cleaning required.
0101h - 7FFFh	Reserved.
8000h - FFFh	Vendor-specific parameters.

A non-zero value of the cleaning required parameter specifies a condition requiring cleaning has been detected and a subsequent cleaning cycle has not been completed. The cleaning required parameter shall be persistent across I_T nexus losses, logical unit resets, and power cycles.

8.2.3 TapeAlert log page

The TapeAlert log page (see table 53) defines error and informational flags used for detailed device diagnostics. Refer to table A.1 (see Annex A) for a description of the flags. Refer to SPC-3 for a definition of the log page fields.

Table 53 — TapeAlert log page

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved		PAGE CODE (2Eh)					
1	Reserved							
2	(MSB)		PARAMETER LENGTH (140h)					
3								(LSB)
5*n*-1	(MSB)		PARAMETER CODE (*n*)					
5*n*								(LSB)
5*n*+1	DU(0)	DS(1)	TSD(0)	ETC(0)	TMC(00b)		LBIN(0)	LP(0)
5*n*+2	PARAMETER LENGTH (1)							
5*n*+3	VALUE OF FLAG							

The PARAMETER LENGTH field shall be set to 140h to allow for the transfer of all 64 flags.

The value of *n* in the PARAMETER CODE field shall be in the range of 1 to 64.

A value of one in the VALUE OF FLAG field specifies the flag is set. Any other value specifies the flag is not set.

8.3 Mode parameters

8.3.1 Mode parameters overview

This subclause defines the descriptors and pages for mode parameters used with sequential-access devices.

The mode parameter list, including the mode parameter header and mode block descriptor, are described in SPC-3.

The MEDIUM-TYPE CODE field in the mode parameter header is vendor-specific for sequential-access devices.

The value of the BLOCK LENGTH field in the mode parameter block descriptor shall be a multiple of four.

> NOTE 41 The block length field is limited to multiples of four to ensure data integrity will be maintained when fixed-block transfers are performed using transports such as Fibre Channel.

The DEVICE-SPECIFIC PARAMETER field in the mode parameter header is defined in table 54 for sequential-access devices.

Table 54 — Device-specific parameter

Bit Byte	7	6	5	4	3	2	1	0
	WP	BUFFERED MODE			SPEED			

When used with the MODE SENSE command, a write protection (WP) bit of zero specifies the medium is write enabled. A WP bit of one specifies the medium is currently in the write protected state. When used with the MODE SELECT command, this field is ignored.

> NOTE 42 The write protected state may be due to logical unit internal restrictions, software write protection, or physical write protection.

Values for the BUFFERED MODE field are defined in table 55.

Table 55 — Buffered modes

Code	Description
0h	The device server shall not report GOOD status on WRITE commands until the logical blocks are actually written on the medium.
1h	The device server may report GOOD status on WRITE commands as soon as all the data specified in the WRITE command has been transferred to the logical unit's object buffer. One or more logical blocks may be buffered prior to writing the logical block(s) to the medium.
2h	The device server may report GOOD status on WRITE commands as soon as: a) all the data specified in the write command has been successfully transferred to the logical unit's object buffer; and b) all buffered logical objects from different initiators has been successfully written to the medium.
3h - 7h	Reserved

Values for the SPEED field shall be assigned as defined in table 56.

Table 56 — SPEED field definition

Code	Description
0h	Default (use the device's default speed).
1h	Use the device's lowest speed.
2h - Fh	Use increasing device speeds.

For the MODE SELECT command, the DENSITY CODE field of the sequential-access device block descriptor (see SPC-3) specifies the density selected by the application client for use in subsequent read and write operations. For logical units capable of automatic density recognition, the density code selected by the application client may be overridden by the logical unit for a subsequent read operation if the selected value does not match the current recorded density of the medium. If the MODE SELECT command specifies the default density code the logical unit selects the actual density code to be used in a vendor-specific manner. The value is expected to be the principal density code (or an optimal density code).

For the MODE SENSE command, the DENSITY CODE field reflects the current operating density of the logical unit. If a current operating density has not been selected, either because no medium is mounted or because the density of the installed medium has not been determined, the DENSITY CODE field should be set to the principal density code value (see 3.1.49). For some logical units, the principal density code value returned in response to a MODE SENSE command may change dynamically to match the most recently detected density. The DENSITY CODE value returned in response to a MODE SENSE command shall be determined as follows:

  a) following a logical unit reset, if the logical unit is not ready, the device server shall report the principal density;
  b) following a unit attention condition for a not-ready-to-ready transition or an unsuccessful read operation, the device server shall:
     A) report the principal density if no attempt has been made by the logical unit to determine the density;
     B) report the principal density if the logical unit is unable to automatically determine the density from the medium; or
     C) report the current medium density if the logical unit has determined the density from the medium.
  c) following a successful read operation, the device server shall report a density code value reflecting the recorded density of the medium. For some implementations, the logical unit may automatically determine this value from the medium. For devices not capable of automatic density determination, the principal density is reported if the density code value is not provided by the preceding MODE SELECT command;
  d) following a successful write operation, the device server shall report a density code value reflecting the most recently recorded density of the medium;
  e) following an unsuccessful read operation or a successful write operation, while at beginning-of-partition, the device server shall report a density code value as described for item b);
  f) following a successful unload operation, the device server shall report the most recent density code value as determined by items b) through e) above; or
  g) following a logical unit reset, if the logical unit is ready, the device server shall retain knowledge of the density code as determined by items b) through e) above.

For a MODE SELECT command, a density code of 7Fh specifies the application client is not selecting a density. The value 7Fh shall not be returned by a MODE SENSE command. Table 57 lists the sequential-access device density codes.

Table 57 — Sequential-access density codes

Code value	Description	Note
00h	Default density.	a
01h - 7Eh	Density code from REPORT DENSITY SUPPORT command.	
7Fh	No change from previous density (NO-OP).	b
80h - FFh	Density code from REPORT DENSITY SUPPORT command.	
a. Only reported by MODE SENSE commands if primary density code for the density. b. This density code value is defined for the MODE SELECT command and shall not be returned by the MODE SENSE command.		

The mode page codes and subpage codes for sequential-access devices are defined in table 58. All page code and subpage code combinations not shown in table 58 are reserved.

Table 58 — Mode page codes and subpage codes

Page code	Subpage code	Mode page name	Reference
00h	not applicable	Vendor-specific (does not require page format)	
01h	00h	Read-Write Error Recovery	8.3.5
02h	00h	Disconnect-Reconnect	SPC-3
03h - 08h	00h - FEh	Reserved	
09h	00h	Obsolete	3.3.7
0Ah	00h	Control	SPC-3
0Ah	01h	Control Extension	SPC-3
0Bh - 0Eh	00h - FEh	Reserved	
0Fh	00h	Data Compression	8.3.2
10h	00h	Device Configuration	8.3.3
11h	00h	Medium Partition	8.3.4
12h	00h	Obsolete	3.3.7
13h	00h	Obsolete	3.3.7
14h	00h	Obsolete	3.3.7
15h	00h	Extended	SPC-3
16h	00h	Extended Device-Type Specific	SPC-3
17h	00h - FEh	Reserved	
18h	00h	Protocol Specific LUN	SPC-3
19h	00h	Protocol Specific Port	SPC-3
1Ah	00h	Power Condition	SPC-3
1Bh	00h - FEh	Reserved	
1Ch	00h	Informational Exceptions Control	8.3.6
1Dh	00h	Medium Configuration	8.3.7
1Dh - 1Fh	00h - FEh	Reserved	
20h - 3Eh	00h - FEh	Vendor-specific (does not require page format)	
3Fh	00h	Return all pages[a]	
3Fh	FFh	Return all pages and subpages[a]	
00h - 3Eh	FFh	Return all subpages[a]	
a) Valid only for MODE SENSE command.			

8.3.2 Data Compression mode page

The Data Compression mode page (see table 59) specifies the parameters for the control of data compression in a sequential-access device.

Table 59 — Data Compression mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(0)	PAGE CODE (0Fh)					
1	PAGE LENGTH (0Eh)							
2	DCE	DCC	Reserved					
3	DDE	RED		Reserved				
4	(MSB)							
5				COMPRESSION ALGORITHM				
6								
7								(LSB)
8	(MSB)							
9				DECOMPRESSION ALGORITHM				
10								
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	Reserved							

See SPC-3 for a description of the PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field.

A data compression enable (DCE) bit of one specifies data compression is enabled. When this bit is one, data sent to the device server by the application client shall be processed using the selected compression algorithm before being written to the medium. A DCE bit of zero specifies data compression is disabled.

A data compression capable (DCC) bit of one specifies the device supports data compression and is capable of processing data sent to it for transferal to the medium using the selected compression algorithm. A DCC bit of zero specifies the device does not support data compression. This shall be a non-changeable bit.

A data decompression enable (DDE) bit of one specifies data decompression is enabled. A DDE bit of zero specifies data decompression is disabled. Uncompressed data shall be unaffected by the setting of the DDE bit.

The report exception on decompression (RED) field specifies the response to certain boundaries detected in the data on the medium. There are a number of boundaries that may occur on the medium between compressed and

uncompressed data. These boundaries are shown in table 60. Only boundaries shown in table 60 may generate a CHECK CONDITION status.

Table 60 — Possible boundaries and resulting sense keys due to data compression

Prior Data	Current Data	Sense Key[a,b]		
		RED **field value**		
		zero	**one**	**two**
Uncompressed	Compressed (unsupported algorithm)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
Uncompressed	Compressed (supported algorithm)	[none]	[none]	RECOVERED ERROR
Compressed (supported algorithm)	Uncompressed	[none]	[none]	NO SENSE
Compressed (supported algorithm)	Compressed (unsupported algorithm)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
Compressed (supported algorithm A)	Compressed (supported algorithm B)	[none]	[none]	RECOVERED ERROR
Compressed (unsupported algorithm)	Uncompressed	[none]	NO SENSE	NO SENSE
Compressed (unsupported algorithm)	Compressed (supported algorithm)	[none]	RECOVERED ERROR	RECOVERED ERROR
Compressed (unsupported algorithm A)	Compressed (unsupported algorithm B)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
All other combinations		[none]	[none]	[none]
a. [none] specifies no CHECK CONDITION status is returned given the data boundary condition and the current value of the RED field. b. The appropriate additional sense code is specified following this table in this subclause.				

If a CHECK CONDITION status is returned and the current data is compressed, the additional sense code shall be set to either DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN with the additional sense code qualifier set to the algorithm id or DECOMPRESSION EXCEPTION LONG ALGORITHM with no additional sense code qualifier.

If a CHECK CONDITION status is returned and the current data is uncompressed, the additional sense code shall be set to DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN with the additional sense code qualifier set to zero.

A RED field of zero specifies the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation that the device is unable to decompress. Data boundaries in table 60 marked other than [none] in the column for RED field values of zero shall generate CHECK CONDITION status with the specified sense key when the RED field is zero.

A RED field of one specifies the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation that requires different handling by the application client than the data most recently encountered during a prior read operation. At each of these boundaries, the data that is sent to the application client is of a fundamentally different nature from that which was previously sent. Data boundaries in table 60

marked other than [none] in the column for RED field values of one shall generate CHECK CONDITION status with the specified sense key when the RED field is one.

A RED field of two specifies the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation that has been processed using a different algorithm from that data most recently encountered during a prior read operation. Data boundaries in table 60 marked other than [none] in the column for RED field values of two shall generate CHECK CONDITION status with the specified sense key when the RED field is two.

A RED field of three is reserved. If a mode page containing a RED field of three is received, the MODE SELECT command shall be terminated with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

Upon detection of any of the boundary conditions described in table 60 that results in a CHECK CONDITION status, the additional sense code shall be set to either DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN (if the algorithm identifier is less than or equal to 255) or DECOMPRESSION EXCEPTION LONG ALGORITHM ID. The device shall, in both cases, set the DECOMPRESSION ALGORITHM field to the algorithm identifier of the compression algorithm used to process the encountered data. The logical position shall be on the EOP side of the encountered data, and the INFORMATION field in the sense data shall contain a count of the number of logical blocks contained within the encountered data.

When compressed data is encountered on the medium that the device server is unable to decompress, the device server shall return a CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR and the additional sense code shall be set to CANNOT DECOMPRESS USING DECLARED ALGORITHM. Undecompressed data may be returned to the application client as a single variable length logical block with the ILI bit and INFORMATION fields set accordingly. The logical position is vendor-specific following this condition.

> NOTE 43 The undecompressed data may contain more than one logical object. As such, the application client should issue a READ POSTION command following this condition to re-establish positioning.

The COMPRESSION ALGORITHM field specifies the currently selected compression algorithm. The default value of the COMPRESSION ALGORITHM field shall specify the default compression algorithm for the device. The field specifies the compression algorithm the device shall use to process data sent to it by the application client when the DCE bit is set to one. If the application client selects an algorithm that the device does not support, then the device shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST. Algorithm identifiers are shown in table 61. The SELECT DATA COMPRESSION ALGORITHM field in the Device Configuration mode page shall be ignored if a Data Compression mode page with the DCE bit set to one is also received by the device in the same MODE SELECT command.

For the MODE SELECT command, the DECOMPRESSION ALGORITHM field specifies the decompression algorithm selected by the application client for use in subsequent decompression of data encountered on the medium. For devices capable of the automatic recognition of the compression algorithm used to process data encountered on the medium, the decompression algorithm selected by the application client may be ignored, or overridden by the logical unit for a subsequent read operation if the selected value does not match the compression algorithm that was used to process the data encountered on the medium.

For the MODE SENSE command, the DECOMPRESSION ALGORITHM field reflects the algorithm selected by the application client. For some devices, the DECOMPRESSION ALGORITHM value returned in response to a MODE SENSE command may change dynamically to match the compression algorithm, detected by the device, that was used to process the data most recently encountered on the medium, during a read operation. A value of zero specifies the

data encountered on the medium during the most recent read operation was uncompressed. Compression algorithm identifiers are shown in table 61.

Table 61 — Compression algorithm identifiers

Algorithm Identifier	Description
00h	No algorithm selected (identifies uncompressed data).
01h	Set with MODE SELECT to select the default algorithm. MODE SENSE shall return the actual compression algorithm that was selected.
02h	Reserved.
03h	IBM ALDC[a] data compression algorithm with 512 byte buffer.
04h	IBM ALDC[a] data compression algorithm with 1024 byte buffer.
05h	IBM ALDC[a] data compression algorithm with 2048 byte buffer.
06h - 0Fh	Reserved.
10h	IBM IDRC[b] data compaction algorithm.
11h - 1Fh	Reserved.
20h	DCLZ[c] data compression algorithm.
21h - FEh	Reserved.
FFh	Unregistered algorithm.
100h - FFFFFFFFh	Reserved.
a. Adaptive Lossless Data Compression (see ISO/IEC 15200:1996). b. Improved Data Recording Capability c. Data Compression according to Lempel and Ziv (see ISO/IEC 11558:1992).	

8.3.3 Device Configuration mode page

The Device Configuration mode page (see table 62) is used to specify the appropriate sequential-access device configuration.

Table 62 — Device Configuration mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(0)	PAGE CODE (10h)					
1	PAGE LENGTH (0Eh)							
2	Rsvd	Obsolete	CAF	ACTIVE FORMAT				
3	ACTIVE PARTITION							
4	WRITE OBJECT BUFFER FULL RATIO							
5	READ OBJECT BUFFER EMPTY RATIO							
6	(MSB)	WRITE DELAY TIME						
7								(LSB)
8	OBR	LOIS	Obsolete	AVC	SOCF		ROBO	REW
9	Obsolete							
10	EOD DEFINED			EEG	SEW	SWP	BAML	BAM
11	(MSB)							
12	OBJECT BUFFER SIZE AT EARLY WARNING							
13								(LSB)
14	SELECT DATA COMPRESSION ALGORITHM							
15	WTRE		OIR	REWIND ON RESET		ASOCWP	PERSWP	PRMWP

See SPC-3 for a description of the PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field.

> NOTE 44 The change active partition (CAP) bit (byte 2, bit 6 in the Device Configuration mode page) has been obsoleted. To change active partitions refer to the LOCATE command.

A change active format (CAF) bit of one specifies the active format is to be changed to the value specified in the ACTIVE FORMAT field. A CAF bit of zero specifies no active format change is specified. For some devices, the format may only be changed when the logical unit is at beginning-of-partition.

The ACTIVE FORMAT field specifies the recording format that is in use for the selected density code when reading or writing data on a logical unit. The value of the ACTIVE FORMAT field is vendor-specific.

The ACTIVE PARTITION field specifies the current logical partition number in use on the medium. This shall be a non-changeable field.

The WRITE OBJECT BUFFER FULL RATIO field, on WRITE commands, specifies to the device server how full the object buffer shall be before writing data to the medium. A value of zero specifies the value is not specified.

The READ OBJECT BUFFER EMPTY RATIO field, on READ commands, specifies to the device server how empty the object buffer shall be before retrieving additional data from the medium. A value of zero specifies the value is not specified.

The WRITE DELAY TIME field specifies the maximum time, in 100 ms increments, that the device server should wait before any buffered data that is to be written, is forced to the medium after the last buffered WRITE command that did not cause the object buffer to exceed the write object buffer full ratio. A value of zero specifies the device server shall never force buffered data to the medium under these conditions.

An object buffer recovery (OBR) bit of one specifies the logical unit supports object buffer recovery using the RECOVER BUFFERED DATA command. An OBR bit of zero specifies the logical unit does not support object buffer recovery. Most device servers consider this bit to be not changeable.

A logical object identifiers supported (LOIS) bit of zero specifies logical object identifiers are not supported in the format written on the medium. A LOIS bit of one specifies the format on the medium has recorded information about the logical object identifiers relative to a partition. Most device servers consider this bit to be not changeable.

The automatic velocity control (AVC) bit of one, specifies the device shall select the speed (if the device supports more than one speed) based on the data transfer rate that should optimize streaming activity and minimize medium repositioning. An AVC bit of zero specifies the speed chosen shall be defined by the SPEED field in the mode parameter header.

A stop on consecutive filemarks (SOCF) field of 00b specifies the device server shall pre-read data from the medium to the limits of the object buffer capacity without regard for filemarks. Values 01b, 10b, and 11b specify that the device server shall terminate the pre-read operation if one, two, or three consecutive filemarks are detected, respectively.

A recover object buffer order (ROBO) bit of one specifies logical blocks shall be returned from the object buffer of the logical unit on a RECOVER BUFFERED DATA command in LIFO order (last-in-first-out) from that they were written to the object buffer. A RBO bit of zero specifies logical blocks shall be returned in FIFO (first-in-first-out) order.

A report early-warning (REW) bit of zero specifies the device server shall not report the early-warning condition for read operations and it shall report early-warning at or before any medium-defined early-warning position during write operations.

A REW bit of one specifies the device server shall return CHECK CONDITION status. The additional sense code shall be set to END-OF-PARTITION/MEDIUM DETECTED, and the EOM bit set to one in the sense data when the early-warning position is encountered during read and write operations. If the REW bit is one and the SEW bit is zero, the device server shall return CHECK CONDITION status with the sense key set to VOLUME OVERFLOW when early-warning is encountered during write operations.

> NOTE 45 A REW bit of one is intended for compatibility with application clients using old tape formats that require an early-warning indication during read operations. Other application clients should set this bit to zero to avoid potential data loss when interchanging tapes between devices.

The EOD DEFINED field specifies the format type that the logical unit shall use to detect and generate the EOD area. The values for EOD DEFINED are specified in table 63.

Table 63 — EOD DEFINED values

Code	Description
000b	Logical unit's default EOD definition
001b	Format-defined erased area of medium
010b	As specified in the SOCF field
011b	EOD recognition and generation is not supported
100b - 111b	Reserved

An enable EOD generation (EEG) bit set to one specifies the logical unit shall generate the appropriate EOD area, as determined by the EOD field. A value of zero specifies EOD generation is disabled.

> NOTE 46 Some logical units may not generate EOD at the completion of any write-type operation.

A synchronize at early-warning (SEW) bit set to one specifies the logical unit shall cause any buffered logical objects to be transferred to the medium prior to returning status when positioned between early-warning and EOP. A SEW bit of zero specifies the logical unit may retain unwritten buffered logical objects in the object buffer when positioned between early-warning and EOP (see 5.6, 5.7, 6.8, and 6.9).

A software write protection (SWP) bit set to one specifies the device server shall perform a synchronize operation then enter the write-protected state (see 4.2.11 and 4.2.11.3). When the SWP bit is set to one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code should be set to LOGICAL UNIT SOFTWARE WRITE PROTECTED (see 4.2.11.2). A SWP bit set to zero specifies the device server may inhibit writing to the medium, dependent on other write inhibits.

A block address mode lock (BAML) bit of zero specifies the selection of the block address mode shall be determined based on the first block address mode unique command that is received after a successful load operation or a successful completion of a command that positions the medium to BOP. A BAML bit of one specifies the selection of the block address mode shall be determined based on the setting of the BAM bit. See 4.2.14 for a description of block address mode selection.

The block address mode (BAM) bit is valid only if the BAML bit is set to one. If the BAML bit is set to zero, the BAM bit shall be ignored. If the BAML bit is set to one and the BAM bit is set to zero, the logical unit shall operate using implicit address mode. If the BAML bit is set to one and the BAM bit is set to one, the logical unit shall operate using explicit address mode. See 4.2.14 for a description of block address mode selection.

The OBJECT BUFFER SIZE AT EARLY WARNING field specifies the value, in bytes, that the logical unit shall reduce its logical object buffer size to when writing in a position between its early-warning and end-of-partition. A value of zero specifies the implementation of this function is vendor-specific.

> NOTE 47 The intent is to prevent the loss of data by limiting the size of the object buffer when near the end-of-partition.

The SELECT DATA COMPRESSION ALGORITHM field set to 00h specifies the logical unit shall not use a compression algorithm on any data sent to it prior to writing the data to the medium. A value of 01h specifies the data to be written shall be compressed using the logical unit's default compression algorithm. Values 02h through 7Fh are reserved. Values 80h through FFh are vendor-specific. The SELECT DATA COMPRESSION ALGORITHM field shall be ignored if a Data Compression mode page with the DCE bit set to one is received by the device in the same MODE SELECT command.

> NOTE 48 New implementations use the Data Compression mode page (see 8.3.2) for specifying data compression behavior.

The WORM Tamper Read Enable (WRTE) field specifies how the device server responds to detection of comprimised integrity of a WORM medium when processing a locate, read, read reverse, space, or verify operation. The

wrte field shall have no effect on the processing of a locate, read, read reverse, space, or verify operation when the device contains a non-WORM medium. The values for the WRTE field are specified in table 64.

Table 64 — WRTE field definition

Code	Description
00b	The device server shall respond in a vendor-specific manner.
01b	Detection of comprimised integrity on a WORM medium shall not affect processing of a task.
10b	The device server shall return CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR and the additional sense code shall be set to WORM MEDIUM - INTEGRITY CHECK. The position of the medium may have changed,
11b	Reserved. The device server shall return CHECK CONDITION status for a MODE SELECT command with the WRTE field set to 11b. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

NOTE 49 An application client should set the WRTE field to 01b only for the recovery of data from a WORM medium where the integrity of the stored data has been comprimised.

If the only if reserved (OIR) bit is set to one, the device server shall process a command only if a reservation (see SPC-2) or persistent reservation (see SPC-3) exists that allows access via the I_T nexus from which the command was received. If the OIR bit is set to one and a command is received from an I_T nexus for which no reservation exists, the device server shall not process the command. If the OIR bit is set to one and a command is received from an I_T nexus for a logical unit upon which no reservation or persistent reservation exists, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to NOT RESERVED. Commands that shall not be effected by the OIR bit set to one are defined as Allowed in the presence of persistent reservations in table 11 or SPC-3, or are defined in SPC-2 as Allowed in the presence of reservations. Commands that shall be effected by the OIR bit set to one are defined as Conflict in the presence of persistent reservations in table 11 or SPC-3, except for the RESERVE, RELEASE, PERSISTENT RESERVATION IN and PERSISTENT RESERVATION OUT commands, or are defined in SPC-2 as Conflict in the presence of reservations. An OIR bit set to zero specifies the device server shall process commands as specified in table 11 or SPC-3.

The REWIND ON RESET field is specified in table 65. The REWIND ON RESET field, if implemented, shall be persistent across logical unit resets.

Table 65 — REWIND ON RESET field definition

Code	Description
00b	Vendor-specific
01b	The logical unit shall position to the beginning of the default data partition (BOP 0) on logical unit reset.
10b	The logical unit shall maintain its position on logical unit reset.
11b	Reserved

An associated write protection (ASOCWP) bit set to one specifies the logical unit shall inhibit all writing to the medium after performing a synchronize operation (see 4.2.11 and 4.2.11.4). When the ASOCWP bit is set to one, the currently mounted volume is logically write protected until the volume is de-mounted (see 4.2.11 and 4.2.11.4). When the ASOCWP bit is set to one, all commands requiring eventual writes to the medium shall return CHECK

CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code should be set to ASSOCIATED WRITE PROTECT (see 4.2.11.2). An ASOCWP bit set to zero specifies the currently mounted volume is not write protected by the associated write protection. The ASOCWP bit shall be set to zero by the device server when the volume is de-mounted. This change of state shall not cause a unit attention condition. If the application client sets the ASOCWP bit to one while no volume is mounted, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense code shall be set to MEDIUM NOT PRESENT. If the Device Configuration mode page is savable, the ASOCWP bit shall be saved as zero, regardless of the current setting.

A persistent write protection (PERSWP) bit set to one specifies the currently mounted volume is logically write protected (see 4.2.11 and 4.2.11.5). When the PERSWP bit is set to one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code should be set to PERSISTENT WRITE PROTECT (see 4.2.11.2). A PERSWP bit set to zero specifies the currently mounted volume is not write protected by the persistent write protection. The PERSWP bit shall be set to zero by the device server when the volume is de-mounted or when a volume is mounted with persistent write protection disabled. The PERSWP bit shall be set to one by the device server when a volume is mounted with persistent write protection enabled. These changes of state shall not cause a unit attention condition. If the application client sets the PERSWP bit to one while no volume is mounted, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense code shall be set to MEDIUM NOT PRESENT. If the application client sets the PERSWP bit to one when the logical position is not at BOP 0, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to POSITION PAST BEGINNING OF MEDIUM. If the Device Configuration mode page is savable, the PERSWP bit shall be saved as zero, regardless of the current setting.

A permanent write protection (PRMWP) bit set to one specifies the currently mounted volume is logically write protected (see 4.2.11 and 4.2.11.6). When the PRMWP bit is set to one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code should be set to PERMANENT WRITE PROTECT (see 4.2.11.2). A PRMWP bit set to zero specifies the currently mounted volume is not write protected by the permanent write protection. The PRMWP bit shall be set to zero by the device server when the volume is de-mounted or when a volume is mounted with permanent write protection disabled. The PRMWP bit shall be set to one by the device server when a volume is mounted with permanent write protection enabled. These changes of state shall not cause a unit attention condition. If the application client sets the PRMWP bit to one while no volume is mounted, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense code shall be set to MEDIUM NOT PRESENT. If the application client sets the PRMWP bit to one when the logical position is not at BOP 0, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to POSITION PAST BEGINNING OF MEDIUM. If the application client attempts to change the PRMWP bit from one to zero, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to DATA PROTECT and the additional sense code shall be set to PERMANENT WRITE PROTECT. If the Device Configuration mode page is savable, the PRMWP bit shall be saved as zero, regardless of the current setting.

8.3.4 Medium Partition mode page

The Medium Partition mode page (see table 66) is used to specify the group of medium partitions. Fields in the Medium Partition mode page indicating the current state of the partitions for the medium shall be changed by the device server to the current medium state on a not ready to ready transition when the medium state changes from de-mounted to mounted. The physical placement and order of medium partitions are not specified by this standard.

> NOTE 50 Since defining partitions may require reformatting the medium for some implementations, an implicit write to the medium may occur as a result of a MODE SELECT command that supplies these parameters.

Table 66 — Medium Partition mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(0)	PAGE CODE (11h)					
1	PAGE LENGTH							
2	MAXIMUM ADDITIONAL PARTITIONS							
3	ADDITIONAL PARTITIONS DEFINED							
4	FDP	SDP	IDP	PSUM		POFM	CLEAR	ADDP
5	MEDIUM FORMAT RECOGNITION							
6	Reserved				PARTITION UNITS			
7	Reserved							
Partition size descriptor(s)								
8	(MSB)	PARTITION SIZE						
9								(LSB)
n-1	(MSB)	PARTITION SIZE						
n								(LSB)

See SPC-3 for a description of the PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field.

The MAXIMUM ADDITIONAL PARTITIONS field is a logical unit-defined value indicating the maximum number of additional partitions supported by the logical unit. A value of zero returned by the MODE SENSE command specifies no additional partitions are present or allowed.

The ADDITIONAL PARTITIONS DEFINED field specifies the number of additional partitions to be defined for a volume when the SDP or IDP bit is set to one. The maximum value allowed is the value returned in the MAXIMUM ADDITIONAL PARTITIONS field. The ADDITIONAL PARTITIONS DEFINED value returned by the MODE SENSE command shall report one less than the number of partitions on the media when the logical unit is ready. If the unit is not ready, the ADDITIONAL PARTITIONS DEFINED field is undefined.

A fixed data partitions (FDP) bit of one specifies the logical unit shall partition the medium based on its fixed definition of partitions if the POFM bit is set to zero. Setting this bit to one when POFM is set to zero may only be valid at beginning-of-partition and is mutually exclusive with the SDP and IDP bits. The partition size descriptors are ignored by the MODE SELECT command when the FDP bit is set to one. The logical unit may assign any number of partitions from 1 to (MAXIMUM ADDITIONAL PARTITIONS + 1).

> NOTE 51 It is recommended that the partition size descriptors be present in MODE SENSE data regardless of the settings of the FDP, SDP or IDP fields to give an estimate of the size of each partition.

A select data partitions (SDP) bit of one specifies the logical unit shall partition the medium into the number of partitions as specified by the ADDITIONAL PARTITIONS DEFINED field (*n*) using partition sizes defined by the device if the POFM bit is set to zero. The logical unit shall partition the medium into *n*+1 partitions numbered 0 through *n*. Setting this bit to one when POFM is set to zero may only be valid at beginning-of-partition and it is mutually exclusive with

the FDP and IDP fields. The partition size descriptors are ignored by the MODE SELECT command when the SDP bit is set to one.

An initiator-defined partitions (IDP) bit of one specifies the logical unit shall partition the medium as defined by the ADDITIONAL PARTITIONS DEFINED field and the partition size descriptors if the POFM bit is set to zero. Setting this bit to one when POFM is set to zero may only be valid at beginning-of-partition and is mutually exclusive with the FDP and SDP fields. The number of non-zero partition size descriptors received in the Medium Partition mode page shall be one more than the ADDITIONAL PARTITIONS DEFINED value. The size of partition 0 shall be non-zero.

A logical unit is not required to retain the method used to partition the medium. The device server shall set only one of the IDP, FDP or SDP fields in the MODE SENSE data. If a volume was previously partitioned through a MODE SELECT command with FDP or SDP set to one, a device server may set IDP to one in subsequent MODE SENSE data since the volume has been initiator partitioned. However, in a MODE SELECT command, the application client cannot use IDP set to one in place of FDP or SDP set to one.

> NOTE 52 Since defining partitions may require reformatting the medium for some implementations, an implicit write to the medium may occur as a result of a MODE SELECT command that has any of the fields FDP, SDP, or IDP set to one and has a value of zero in the POFM field."

The partition size unit of measure (PSUM) field defines the units of the partition size descriptors. A logical unit is not required to retain the partition size unit of measure used to partition the medium. The PSUM field is defined in table 67.

Table 67 — PSUM values

Code	Description	Support
00b	bytes (unit of one)	O
01b	kilobytes (10^3 bytes)	O
10b	megabytes (10^6 bytes)	O
11b	$10^{(\text{PARTITION UNITS})}$ bytes	O

The PARTITION UNITS field defines the size of the partition size descriptors when the PSUM field is set to 11b. A value of n in the PARTITION UNITS field shall define the units of the partition size descriptors as 10^n bytes. If the PARTITION UNITS field is supported, all possible values shall be supported. A logical unit is not required to retain the partition units used to partition the medium. If PSUM is not equal to 11b, the PARTITION UNITS field is undefined. Some values of the PARTITION UNITS field may result in no legal non-zero partition size descriptors.

A partition on format (POFM) bit of one specifies the MODE SELECT command shall not cause changes to the partition sizes or user data, either recorded or buffered. If POFM is set to one, actual media partitioning occurs when the device server receives a subsequent FORMAT MEDIUM command (see 7.1). When the FORMAT MEDIUM command partitions the media, it shall do so based on the contents of the mode data for the Medium Partition mode page. If POFM is set to one, field values specified by a MODE SELECT command for the Medium Partition mode page shall not be changed by the device server before the media is unloaded or until a logical unit reset. Some field checking may be performed by the MODE SELECT command. However, there is no guarantee that any subsequent partitioning during a FORMAT MEDIUM command will complete with no errors.

A POFM bit of zero specifies the MODE SELECT command shall alter the partition information for the medium if any of the SDP, FDP, or IDP bits are set to one.

A CLEAR bit of zero and an ADDP bit of zero specifies the logical unit may logically erase any or all partitions when one of the IDP, FDP, or SDP fields are set to one by a MODE SELECT command.

A CLEAR bit of one and an ADDP bit of zero specifies the logical unit shall logically erase every partition if one of the IDP, FDP, or SDP fields is set to one. No formatting of the medium is implied.

An ADDP bit of one and a CLEAR bit of zero specifies the logical unit shall not logically erase any existing partitions, even if the size of the partition is changed. If the MODE SELECT command partition size descriptor and the current partition size differ, the logical unit shall truncate or extend the partition, whichever is appropriate. If the MODE SELECT command partition size is zero and the current partition size is non-zero, the partition shall be logically removed from the medium, resulting in the loss of all data in that partition. If the MODE SELECT command partition size is equivalent to the current partition size, no change in the partition size shall result. If the logical unit is unable to perform the operation or if such an operation would cause loss of valid data in any partition that exists both before and after the MODE SELECT or FORMAT MEDIUM command, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the addition sense code set to PARAMETER VALUE INVALID. If the ADDP bit is set to one and either ADDP is not supported or the FDP field is set to one the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST. If both the ADDP and SDP fields are set to one, the logical unit shall add or remove partitions such that the resulting partition count on the medium is equal to the ADDITIONAL PARTITIONS DEFINED value plus one.

If both the ADDP and CLEAR fields are set to one, the logical unit shall logically erase all partitions that differ in size from the corresponding partition size descriptor in the MODE SELECT data. Partitions with the same size as the MODE SELECT data size shall retain all existing data. If the logical unit is incapable of supporting the changes requested without loss of data, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER VALUE INVALID. If setting both ADDP and CLEAR to one is not supported, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

A MODE SELECT command partition size descriptor has the equivalent (same) size as the current partition size if:

   a) the mode select PARTITION SIZE, PSUM, and PARTITION UNITS fields are exactly the same as those returned by MODE SENSE command;
   b) the mode select PARTITION SIZE field value is within plus or minus one of the current size when the current size is converted to the units of the mode select PSUM or PARTITION UNITS field; or
   c) the mode select PARTITION SIZE is FFFFh and the current size would return FFFFh if expressed in the units of the mode select PSUM or PARTITION UNITS field.

The MEDIUM FORMAT RECOGNITION field specifies the logical unit's capability to automatically identify the medium format and partition information when reading a volume. The value in this field may be different following a medium change. The MEDIUM FORMAT RECOGNITION field values are shown in table 68.

Table 68 — Medium format recognition values

Code	Description
00h	Logical unit is incapable of format or partition recognition.
01h	Logical unit is capable of format recognition.
02h	Logical unit is capable of partition recognition.
03h	Logical unit is capable of format and partition recognition.
04h - FFh	Reserved

NOTE 53 If a logical unit specifies it is not capable of medium format recognition, the application client should supply all necessary parameters for the device to identify the specific format.

PARTITION SIZE fields within the partition size descriptor list define the approximate size of the respective partitions in the units specified in the PSUM and PARTITION UNITS fields. Partitions are numbered by their relative position in the partition size descriptor list, starting at 0. Only partition numbers in the range of 0 to *n* where *n* is less than or equal to 63 may have size descriptors in this mode page. Partition *n*, if present, shall be described by the partition size descriptor at mode page offsets 8+(2∗*n*) and 9+(2∗*n*). Partition 0 shall be the default partition. Partition size descriptor 0, shall contain the size of the default partition. The size of partition 0 shall be greater than 0. Up to 64 partitions may be defined using this mode page. Partitions not assigned shall have a partition size descriptor of 0. The logical unit may support more partitions than partition size descriptors. A logical unit may support more partition size descriptors than supported by the medium. All partition size descriptors representing a partition number greater than the maximum additional partition count shall be 0. The partition size descriptors are undefined if the logical unit is not ready. A MODE SELECT command partition size descriptor of FFFFh requests that the logical unit allocate all remaining partition space to that partition. A MODE SENSE command shall return a partition size descriptor of FFFFh if the partition size, in units of PSUM or PARTITION UNITS, is greater than or equal to FFFFh. If insufficient space exists on the medium for the requested partition sizes or if multiple partition size descriptors are set to FFFFh, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST. A device server may round, as described by the MODE SELECT command in SPC-3, any partition size to the nearest valid partition size.

> NOTE 54 It is recommended, but not required, that the number of partition size descriptors available through the Medium Partition mode page equal at least the number of maximum addition partitions + 1. This provides a mechanism for the device server to disclose the current partition sizes.

8.3.5 Read-Write Error Recovery mode page

The Read-Write Error Recovery mode page (see table 69) specifies the error recovery and reporting parameters that the device server shall use when transferring data between the device and the medium. These parameters do not affect protocol-level recovery procedures or positioning error recovery procedures.

Table 69 — Read-Write Error Recovery mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(0)	PAGE CODE (01h)					
1	PAGE LENGTH (0Ah)							
2	Reserved		TB	Rsvd	EER	PER	DTE	DCR
3	READ RETRY COUNT							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	WRITE RETRY COUNT							
9	Reserved							
10	Reserved							
11	Reserved							

> NOTE 55 The parameters in the Read-Write Error Recovery mode page also apply to verify operations.

See SPC-3 for a description of the PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field.

A transfer block (TB) bit of one specifies the device server shall use its best effort to transfer a logical block that cannot be read successfully within the specified read recovery limits to the application client before CHECK CONDITION status is returned. A TB bit of zero specifies an unrecoverable logical block shall not be transferred to the application client. Logical blocks that are recoverable within the recovery limits are always transferred, regardless of the value of the TB bit.

An enable early recovery (EER) bit of one specifies the logical unit shall use the most expedient error recovery algorithm (e.g., attempt error correction prior to retries). An EER bit of zero specifies the logical unit shall use the most deliberate error recovery algorithm, within the limits established by the other error recovery parameters (e.g., attempt to recover the logical block error-free prior to using error correction).

A post error (PER) bit of one specifies the device server shall return CHECK CONDITION status to report recovered errors. A PER bit of zero specifies the device server shall not report errors recovered within the limits established by the error recovery parameters. If this bit is zero, the DTE bit shall also be set to zero.

A disable transfer on error (DTE) bit of one specifies the device server shall terminate the data transfer after a recovered read or write error occurs. All data from the recovered logical block shall be transferred prior to terminating the read or write operation. A DTE bit of zero specifies the device server shall not terminate the transfer for errors recovered within the limits established by the read-write error recovery parameters.

A disable correction (DCR) bit of one specifies the logical unit shall not use error correction codes during error recovery. A DCR bit of zero allows the use of error correction codes for error recovery.

The READ RETRY COUNT field specifies the number of times that the logical unit should attempt its recovery algorithm during a read operation before an unrecoverable error is reported. A READ RETRY COUNT of zero specifies the logical unit shall not use its recovery algorithm during read operations.

The WRITE RETRY COUNT field specifies the number of times that the logical unit should attempt its recovery algorithm during a write operation before an unrecoverable error is reported. A WRITE RETRY COUNT of zero specifies the logical unit shall not use its recovery algorithm during write operations.

8.3.6 Informational Exceptions Control mode page

The Informational Exceptions Control mode page (see table 70) is used to specify the parameters for the control of TapeAlert specific informational exception conditions for a sequential-access device.

Table 70 — Informational Exceptions Control mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(0)	PAGE CODE (1Ch)					
1	PAGE LENGTH (0Ah)							
2	PERF	Rsvd	EBF	EWASC	DEXCPT	TEST	Rsvd	LOGERR
3	Reserved				MRIE			
4	(MSB)							
5								
6				INTERVAL TIMER				
7								(LSB)
8	(MSB)							
9								
10				REPORT COUNT/TEST FLAG NUMBER				
11								(LSB)

See SPC-3 for a description of the PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field.

See SPC-3 for a description of the LOGERR bit.

A TEST bit of one specifies the target shall generate test/false informational exceptions conditions as follows:

a) if the TEST bit is one and the TEST FLAG NUMBER value is zero, the target shall generate a false informational exception condition based on the MRIE field (the INTERVAL TIMER field is ignored and the REPORT COUNT field is used as the TEST FLAG NUMBER). When a false informational exception condition is posted, the TapeAlert flags in the log page shall not be modified. True informational exception conditions shall have priority over false informational exception conditions. The TEST bit shall be automatically set to zero when the false informational exception condition is posted on the first command (excluding INQUIRY, REPORT LUNS, and REQUEST SENSE commands) that is received with no real informational exception condition pending. The false informational exception condition shall be reported in the method specified by the MRIE value and the additional sense code shall be set to FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE). If the TEST and DEXcpt bits are one and the TEST FLAG NUMBER value is zero, the target shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST; or

b) if the TEST bit is one and the TEST FLAG NUMBER value is a valid non-zero value, the target shall generate or clear a test informational exception condition based on the TEST FLAG NUMBER value as described in table 71.

Table 71 — TapeAlert test descriptions

Value	Description
1 to 64	Activate the TapeAlert flag specified by the TEST FLAG NUMBER field in the log page. Once the TapeAlert flag is activated it shall be processed normally based on the DEXCPT, MRIE, INTERVAL COUNT, and REPORT COUNT values.
-1 to -64	Deactivate the TapeAlert flag specified by the absolute value of the TEST FLAG NUMBER field in the log page. Deactivating the flag in this way is equivalent to performing the specified corrective action for that flag, allowing a true informational exception condition to be activated if the true error condition occurs for that flag.
32767	Activate all of the supported TapeAlert flags in the log page. Once the supported TapeAlert flags are activated, they shall be processed normally based on the DEXCPT, MRIE, INTERVAL COUNT, and REPORT COUNT values.
all others	Reserved.

A TEST bit of zero specifies the target shall not generate any test/false informational exception conditions. The value of the TEST bit returned by a MODE SENSE command shall be zero.

See SPC-3 for a description of the DEXCPT bit, EWASC bit, EBF bit, PERF bit.

See SPC-3 for a description of the MRIE field. If the informational exception condition was generated by an event that caused a real CHECK CONDITION to occur, then this real CHECK CONDITION shall over-ride (i.e., be used instead of) the CHECK CONDITION defined in MRIE modes 01h to 05h. For MRIE modes 01h to 06h, an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED specifies a TapeAlert event has occurred on the device. Detailed information about the event is stored in the TapeAlert log page. Note that the MRIE modes do not effect the logging of events in the TapeAlert log page.

See SPC-3 for a description of the INTERVAL TIMER field.

The REPORT COUNT/TEST FLAG NUMBER field has a dual purpose and is described as follows:

a) if the TEST bit is set to zero, the value of the REPORT COUNT/TEST FLAG NUMBER field represents the REPORT COUNT. The REPORT COUNT value shall be returned in response to a MODE SENSE command. When the REPORT COUNT field is set to zero, this specifies there is no limit on the number of times the target shall report the informational exception condition. When the REPORT COUNT field is not set to zero, this specifies the number of times to report an informational exception condition. If multiple TapeAlert flags are active simultaneously, the flags are reported as a single informational exception condition. The REPORT COUNT is controlled by the newest flag that is active.
b) if the TEST bit is set to one, the value of the REPORT COUNT/TEST FLAG NUMBER field represents the TEST FLAG NUMBER. The TEST FLAG NUMBER value shall not be returned in response to a MODE SENSE command. Valid values of TEST FLAG NUMBER are –64 to 64 and 32767. Negative numbers shall be represented using the 2's complement notation and shall be sign extended to 4 bytes. If the TEST FLAG NUMBER is set to an invalid value, the device server shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST. If the device server does not support a valid TEST FLAG NUMBER and the TEST bit is set to one, the device server shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

8.3.7 Medium Configuration mode page

The Medium Configuration mode page (see table 72) specifies any special considerations the device server shall use when processing commands that access the medium.

Table 72 — Medium Configuration mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF(0)	PAGE CODE (1Dh)					
1	PAGE LENGTH (1Eh)							
2	Reserved							WORMM
3	Reserved							
4	WORM MODE LABEL RESTRICTIONS							
5	WORM MODE FILEMARK RESTRICTIONS							
6-31	Reserved							

See SPC-3 for a description of the PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field.

The WORM mode (WORMM) bit shall be set to one when the device server is operating in WORM mode (see 4.2.18.3). The WORMM bit shall be set to zero when the device server is not operating in WORM mode. If a MODE SELECT command is processed that attempts to change to setting of the WORMM bit, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

The WORM MODE LABEL RESTRICTIONS field specifies the restrictions against overwriting format labels when operating in WORM mode (see table 73). A series of filemarks with no interleaved logical blocks immediately preceding EOD is treated as a filemark sequence and is controlled by the WORM MODE FILEMARKS RESTRICTIONS field.

Table 73 — WORM MODE LABEL RESTRICTIONS field values

Value	Description
00h	The device server shall not allow any logical blocks to be overwritten.
01h	The device server shall not allow some types of format labels to be overwritten.
02h	The device server shall allow all format labels to be overwritten.
03h - FFh	Reserved

The WORM MODE FILEMARKS RESTRICTIONS field specifies the restrictions against overwriting a series of filemarks immediately preceding EOD when operating in WORM mode (see table 74). The WORM MODE FILEMARKS RESTRICTIONS field controls only the overwriting of a series of filemarks with no interleaved logical blocks immediately preceding EOD.

Table 74 — WORM MODE FILEMARKS RESTRICTIONS field values

Value	Description
00h - 01h	Reserved
02h	The device server shall allow any number of filemarks immediately preceding EOD to be overwritten except the filemark closest to BOP.
03h	The device server shall allow any number of filemarks immediately preceding EOD to be overwritten.
04h - FFh	Reserved

8.4 Vital product data (VPD) parameters

8.4.1 VPD parameters overview and page codes

This subclause defines the VPD pages used with sequential-access device types. See SPC-3 for VPD pages used with all device types. The VPD page codes specific to sequential-access devices are specified in table 75.

Table 75 — Sequential-access device VPD page codes

Page code	VPD page name	Reference	Support Requirements
B0h	Sequential-access device capabilities VPD page	8.4.2	Optional
B1h – BFh	Reserved for this device type		

8.4.2 Sequential-access device capabilities VPD page

Table 76 specifies the sequential-access device capabilities VPD page. This page provides the application client with the means to determine if the features specified in this page are supported by the device server.

Table 76 — Sequential-access device capabilities VPD page

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (B0h)							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	Reserved							WORM
5-n	Reserved							

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in SPC-3.

The PAGE LENGTH field specifies the length of the following VPD page data. If the allocation length value in the INQUIRY command descriptor block is too small to transfer all of the VPD page data, the page length shall not be adjusted to reflect the truncation.

If the Write Once Read Many (WORM) bit is set to one, the device server supports WORM mode operation (see 4.2.18.3). If the WORM bit is set to zero, the device server does not support WORM mode operation.

<center>**Annex A**</center>
<center>(normative)</center>

A.1 TapeAlert log page parameter codes (flags)

Table A.1 specifies the TapeAlert log page parameter codes and associated information.

In Table A.1 the following keywords are used:

O = optional

M = mandatory

C = critical

W = warning

I = informational

<center>**Table A.1 — TapeAlert log page parameter codes**</center>

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
01h	Read warning	O	W	The tape drive is having problems reading data. No data has been lost, but there has been a reduction in the performance of the tape.	The drive is having severe trouble reading.
02h	Write warning	O	W	The tape drive is having problems writing data. No data has been lost, but there has been a reduction in the capacity of the tape.	The drive is having severe trouble writing.
03h	Hard error	M	W	The operation has stopped because an error has occurred while reading or writing data that the drive cannot correct.	The drive had a hard read or write error.
04h	Media	M	C	Your data is at risk: 1. Copy any data you require from this tape. 2. Do not use this tape again. 3. Restart the operation with a different tape.	Media can no longer be written/read, or performance is severely degraded.
05h	Read failure	M	C	The tape is damaged or the drive is faulty. Call the tape drive supplier help line.	The drive can no longer read data from the tape.
06h	Write failure	M	C	The tape is from a faulty batch or the tape drive is faulty: 1. Use a good tape to test the drive. 2. If the problem persists, call the tape drive supplier help line.	The drive can no longer write data to the tape.
a) Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consistency to be used to store data (i.e., audio/video grade media should not be used).					

Table A.1 — TapeAlert log page parameter codes (Continued)

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
07h	Media life	O	W	The tape cartridge has reached the end of its calculated useful life: 1. Copy any data you need to another tape. 2. Discard the old tape.	The media has exceeded its specified life.
08h	Not data grade	O	W	The cartridge is not data-grade. Any data you write to the tape is at risk. Replace the cartridge with a data-grade tape.	The drive has not been able to read the MRS[a] stripes.
09h	Write protect	O	C	You are trying to write to a write protected cartridge. Remove the write protection or use another tape.	Write command is attempted to a write protected tape.
0Ah	No removal	O	I	You cannot eject the cartridge because the tape drive is in use. Wait until the operation is complete before ejecting the cartridge.	Manual or software unload attempted when prevent media removal is on.
0Bh	Cleaning media	O	I	The tape in the drive is a cleaning cartridge.	Cleaning tape loaded into drive.
0Ch	Unsupported format	O	I	You have tried to load a cartridge of a type that is not supported by this drive.	Attempted load of unsupported tape format (e.g., DDS2 in DDS1 drive).
0Dh	Recoverable mechanical cartridge failure	O	C	The operation has failed because the tape in the drive has experienced a mechanical failure: 1. Discard the old tape. 2. Restart the operation with a different tape.	Tape snapped/cut or other cartridge mechanical failure in the drive where medium can be de-mounted.
0Eh	Unrecoverable mechanical cartridge failure	O	C	The operation has failed because the tape in the drive has experienced a mechanical failure: 1. Do not attempt to extract the tape cartridge. 2. Call the tape drive supplier help line.	Tape snapped/cut or other cartridge mechanical failure in the drive where medium cannot be de-mounted.
0Fh	Memory chip in cartridge failure	O	W	The memory in the tape cartridge has failed, which reduces performance. Do not use the cartridge for further write operations.	Memory chip failed in cartridge.
a) Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consistency to be used to store data (i.e., audio/video grade media should not be used).						

Table A.1 — TapeAlert log page parameter codes (Continued)

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
10h	Forced eject	O	C	The operation has failed because the tape cartridge was manually de-mounted while the tape drive was actively writing or reading.	Manual or forced eject while drive actively writing or reading.
11h	Read only format	O	W	You have loaded a cartridge of a type that is read-only in this drive. The cartridge will appear as write protected.	Media loaded that is read-only format.
12h	Tape directory corrupted on load	O	W	The tape directory on the tape cartridge has been corrupted. File search performance will be degraded. The tape directory can be rebuilt by reading all the data on the cartridge.	Tape drive powered down with tape loaded, or permanent error prevented the tape directory being updated.
13h	Nearing media life	O	I	The tape cartridge is nearing the end of its calculated life. It is recommended that you: 1. Use another tape cartridge for your next backup. 2. Store this tape cartridge in a safe place in case you need to restore data from it.	Media may have exceeded its specified number of passes.
14h	Clean now	O	C	The tape drive needs cleaning: 1. If the operation has stopped, eject the tape and clean the drive. 2. If the operation has not stopped, wait for it to finish and then clean the drive. Check the tape drive users manual for device specific cleaning instructions.	The drive thinks it has a head clog or needs cleaning.
15h	Clean periodic	O	W	The tape drive is due for routine cleaning: 1. Wait for the current operation to finish. 2. Then use a cleaning cartridge. Check the tape drive users manual for device specific cleaning instructions.	The drive is ready for a periodic cleaning.
16h	Expired cleaning media	O	C	The last cleaning cartridge used in the tape drive has worn out: 1. Discard the worn out cleaning cartridge. 2. Wait for the current operation to finish. 3. Then use a new cleaning cartridge.	The cleaning tape has expired.
17h	Invalid cleaning tape	O	C	The last cleaning cartridge used in the tape drive was an invalid type: 1. Do not use this cleaning cartridge in this drive. 2. Wait for the current operation to finish. 3. Then use a valid cleaning cartridge.	Invalid cleaning tape type used.
a) Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consistency to be used to store data (i.e., audio/video grade media should not be used).					

Table A.1 — TapeAlert log page parameter codes (Continued)

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
18h	Reten-sion requeste d	O	W	The tape drive has requested a retension operation.	The drive is having severe trouble read-ing or writing, that will be resolved by a retension cycle.
19h	Dual-por t inter-face error	O	W	A redundant interface port on the tape drive has failed.	Failure of one inter-face port in a dual-port configura-tion (i.e., Fibre Chan-nel)
1Ah	Cooling fan fail-ure	O	W	A tape drive cooling fan has failed.	Fan failure inside tape drive mecha-nism or tape drive enclosure.
1Bh	Power supply failure	O	W	A redundant power supply has failed inside the tape drive enclosure. Check the enclosure users manual for instructions on replacing the failed power supply.	Redundant PSU fail-ure inside the tape drive enclosure or rack subsystem.
1Ch	Power con-sump-tion	O	W	The tape drive power consumption is outside the specified range.	Power consumption of the tape drive is outside specified range.
1Dh	Drive mainte-nance	O	W	Preventive maintenance of the tape drive is required. Check the tape drive users manual for device specific preventive maintenance tasks or call the tape drive supplier help line.	The drive requires preventive mainte-nance (not cleaning).
1Eh	Hard-ware A	O	C	The tape drive has a hardware fault: 1. Eject the tape or magazine. 2. Reset the drive. 3. Restart the operation.	The drive has a hardware fault that requires reset to recover.
1Fh	Hard-ware B	M	C	The tape drive has a hardware fault: 1. Turn the tape drive off and then on again. 2. Restart the operation. 3. If the problem persists, call the tape drive supplier help line.	The drive has a hardware fault that is not read/write related or requires a power cycle to recover.
20h	Interface	O	W	The tape drive has a problem with the application cli-ent interface: 1. Check the cables and cable connections. 2. Restart the operation.	The drive has identi-fied an interface fault.
a) Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consis-tency to be used to store data (i.e., audio/video grade media should not be used).				

Table A.1 — TapeAlert log page parameter codes (Continued)

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
21h	Eject media	O	C	The operation has failed: 1. Eject the tape or magazine. 2. Insert the tape or magazine again. 3. Restart the operation.	Error recovery action.
22h	Down-load fail	O	W	The firmware download has failed because you have tried to use the incorrect firmware for this tape drive. Obtain the correct firmware and try again.	Firmware download failed.
23h	Drive humidity	O	W	Environmental conditions inside the tape drive are outside the specified humidity range.	Drive humidity limits exceeded.
24h	Drive temper-ature	O	W	Environmental conditions inside the tape drive are outside the specified temperature range.	Cooling problem.
25h	Drive voltage	O	W	The voltage supply to the tape drive is outside the specified range.	Drive voltage limits exceeded.
26h	Predic-tive fail-ure	O	C	A hardware failure of the tape drive is predicted. Call the tape drive supplier help line.	Predictive failure of drive hardware.
27h	Diag-nostics required	O	W	The tape drive may have a hardware fault. Run extended diagnostics to verify and diagnose the problem. Check the tape drive users manual for device specific instructions on running extended diagnostic tests.	The drive may have a hardware fault that may be identified by extended diagnostics (i.e., SEND DIAGNOSTIC command).
28h - 2Eh	Obso-lete				
2Fh - 31h	Rsvd				
32h	Lost sta-tistics	O	W	Media statistics have been lost at some time in the past.	Drive or library powered on with tape loaded.
33h	Tape direc-tory invalid at unload	O	W	The tape directory on the tape cartridge just unloaded has been corrupted. File search performance will be degraded. The tape directory can be rebuilt by reading all the data.	Error preventing the tape directory being updated on unload.
a) Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consistency to be used to store data (i.e., audio/video grade media should not be used).					

Table A.1 — TapeAlert log page parameter codes (Continued)

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
34h	Tape system area write failure	O	C	The tape just unloaded could not write its system area successfully: 1. Copy data to another tape cartridge. 2. Discard the old cartridge.	Write errors while writing the system area on unload.
35h	Tape system area read failure	O	C	The tape system area could not be read successfully at load time: 1. Copy data to another tape cartridge.	Read errors while reading the system area on load.
36h	No start of data	O	C	The start of data could not be found on the tape: 1. Check that you are using the correct format tape. 2. Discard the tape or return the tape to your supplier.	Tape damaged, bulk erased, or incorrect format.
37h	Loading failure	O	C	The operation has failed because the media cannot be loaded and threaded. 1. Remove the cartridge, inspect it as specified in the product manual, and retry the operation. 2. If the problem persists, call the tape drive supplier help line.	The drive is unable to load the media and thread the tape.
38h	Unre- cover- able unload failure	O	C	The operation has failed because the medium cannot be unloaded: 1. Do not attempt to extract the tape cartridge. 2. Call the tape driver supplier help line.	The drive is unable to unload the medium.
39h	Automa- tion inter- face fail- ure	O	C	The tape drive has a problem with the automation interface: 1. Check the power to the automation system. 2. Check the cables and cable connections. 3. Call the supplier help line if problem persists.	The drive has identi- fied an interface fault.
3Ah	Firm- ware failure	O	W	The tape drive has reset itself due to a detected firm- ware fault. If problem persists, call the supplier help line.	Firmware bug.
3Bh	WORM Medium - Integ- rity Check Failed	O	W	The tape drive has detected an inconsistency during the WORM medium integrity checks. Someone may have tampered with the cartridge.	Someone has tam- pered with the WORM medium.
a) Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consis- tency to be used to store data (i.e., audio/video grade media should not be used).					

Table A.1 — TapeAlert log page parameter codes (Continued)

Code	Flag	Type	Flag type	Recommended application client message	Probable cause
3Ch	WORM Medium - Over-write Attempted	O	W	An attempt had been made to overwrite user data on a WORM medium: 1. If a WORM medium was used inadvertently, replace it with a normal data medium. 2. If a WORM medium was used intentionally: a) check that the software application is compatible with the WORM medium format you are using. b) check that the medium is bar-coded correctly for WORM.	The application soft-ware does not recog-nize the medium as WORM.
3Dh	Rsvd				
3Eh	Rsvd				
3Fh	Rsvd				
40h	Rsvd				
a)	Media Recognition System (MRS) is a method where pre-defined stripes are placed at the beginning of the media to identify the media. The MRS stripes are read to determine if the media is of data-grade. Data-grade media should be used in SCSI streaming devices since it is of the required quality and consis-tency to be used to store data (i.e., audio/video grade media should not be used).				