WORKING DRAFT

X3T10 Project 997D

Revision 6
27-DEC-95

Information technology - SCSI-3 Stream Device Command Set (SSC)

This is a draft proposed American National Standard of Accredited Standards Committee X3. As such this is not a completed standard. The X3T10 Technical Committee may modify this document as a result of comments received during the public review process and the approval of this document as a standard. Use of the information contained herein is at your own risk.

Permission is granted to members of X3 and ISO, their technical committees, and their associated task groups to reproduce this document for the purposes of X3 and ISO standardization activities without further permission, provided this notice is included. All other rights are reserved. Any commercial or for-profit use is strictly prohibited.

SSC Technical Editor:

Edward Lappin
Exabyte Corporation
MS 7W2
1685 38th Street
Boulder, CO, 80301

Telephone: 303-417-7718
Facsimile: 303-417-7180
Email: tedl@exabyte.com

ISO LOGO

IEC LOGO

Reference number
ISO/IEC **** : 199x
ANSI X3.*** - 199x
Printed January, 28, 1996 9:32PM

POINTS OF CONTACT:

X3T10 Chair
John B. Lohmeyer
NCR Corporation
1635 Aeroplaza Drive
Colo Spgs, CO 80916

Tel: (719) 573-3362
Fax: (719) 597-3037
Email: john.lohmeyer@symbios.com

X3T10 Vice-Chair
Lawrence J. Lamers
Adaptec
691 South Milpitas Blvd.
Milpitas, CA 95035

Tel: (408) 957-7817
Fax: (408) 957-7193
Email: ljlamers@aol.com

X3 Secretariat
Lynn Barra
Administrator Standards Processing
X3 Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005

Tel: (202) 626-5738
Fax: (202) 638-4922

SCSI Reflector
Internet address for subscription to the SCSI reflector: scsi-request@symbios.com
Internet address for distribution via SCSI reflector: scsi@symbios.com

SCSI Bulletin Board
719-574-0424

Document Distribution
Global Engineering
15 Inverness Way East
Englewood, CO 80112-5704

Tel: (303) 792-2181 or
(800) 854-7179
Fax: (303) 792-2192

ABSTRACT

This standard specifies functional requirements for the SCSI-3 Stream Device Command Set (SSC).

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

PATENT STATEMENT

The developers of this standard have requested that holder's of patents that may be required for the implementation of the standard, disclose such patents to the publisher. However neither the developers nor the publisher have undertaken a patent search in order to identify which if any patents may apply to this standard.

No position is taken with respect to the validity of any claim or any patent rights that may have been disclosed. Details of submitted statements may be obtained from the publisher concerning any statement of patents and willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license.

Document Status

REVISION 0:

- The initial version of SSC.
- Formatting for ISO standards.

REVISION 1:

- Added SCSI-3 approved changes
- Removed processor device model (moved to SPC)
- Changed references from target to logical unit where required
- Elaborated on medium partition pages (1-4).

REVISION 2:

- Reformat to Word (was in Word Perfect 5.1)
- Consistency changes in printer and communications mode pages
- Removed sequential device mode pages from communication mode page list

REVISION 3:

- Mode pages 15h - 1Fh of sequential devices are reserved, not vendor-unique.
- Incorporated 94-152r3 - partition mode pages for sequential devices.
- Incorporated 95-118r1 - Read Position for sequential devices is mandatory; addition of new data bits.
- Incorporated 95-134r0 - Initiator-controlled write protect.
- Other assorted corrected marked with change bars.
- Fixed miscellaneous text problems.

REVISION 4:

- Update contacts section.
- Clarify logical position when CAP is used in the device configuration page of Mode Select.
- Fixed byte number of partition in LOCATE command.
- Fixed error in Mode page 11h IDP bit.
- Fixed description in READ BLOCK LIMITS command.
- Increased side margins.
- Put Prevent/Allow Media removal command back in.
- Do I need a section on reservations???

REVISION 5:

- Removed text for Prevent/Allow media text since the command is back in SPC.
- Changed media type field in mode data to vendor unique.
- Added text indicating that erase, format medium, read reverse, send diagnostics, space, codeload, and format partitions flushes buffered data.
- Added text describing locate to and past end-of-data.
- Added text in model describing identification of blocks, setmarks,and filemark locations.
- Added 95-224r5, report density support. Added note as per working group.
- Moved software write protect, 95-318r1. Function goes better with new page.
- Added 95-304r1 with some minor editorial changes.

REVISION 6:

- Added text for reserve/release for all commands.
- “Samonizing” the document.

- Added text for ready state of printers and communications devices.
- Added definitions and abbreviations.
- Fixed clause 1.
- Added reserve/release to communication device model (mandatory).
- Added text concerning terminated writes and partial logical blocks (see WRITE command).
- Updated drawings for accuracy and style. All drawings altered.
- Added table of notes.

Contents

Page

Forward xi

Introduction xii

1 Scope 1

2 Normative references 3
2.1 Corequisite Standards 3

3 Definitions, symbols and abbreviations 4
3.1 Definitions 4
3.2 Symbols and abbreviations 5

4 General 7
4.1 Overview 7
4.2 Conventions 7

5 Sequential-access devices 9
5.1 Sequential-access device model 9
5.1.1 Physical elements 9
5.1.2 Data storage characteristics 10
5.1.3 Partitions within a volume 12
5.1.4 Logical elements within a partition 13
5.1.5 Data buffering 14
5.1.6 Recorded object descriptors (block identifiers) 15
5.1.7 Direction and position definitions 16
5.1.8 Error reporting 16
5.2 Command descriptions for sequential-access devices 18
5.2.1 ERASE command 19
5.2.2 FORMAT MEDIUM command 19
5.2.3 LOAD UNLOAD command 21
5.2.4 LOCATE command 22
5.2.5 READ command 23
5.2.6 READ BLOCK LIMITS command 25
5.2.7 READ POSITION command 26
5.2.8 READ REVERSE command 29
5.2.9 RECOVER BUFFERED DATA command 30
5.2.10 REPORT DENSITY SUPPORT Command 31
5.2.11 REWIND command 34
5.2.12 SPACE command 35
5.2.13 VERIFY command 37
5.2.14 WRITE command 39
5.2.15 WRITE FILEMARKS command 40
5.3 Parameters for sequential-access devices 41
5.3.1 Diagnostic parameters 42
5.3.2 Log parameters 42
5.3.2.1 Sequential-access device page 42
5.3.3 Mode parameters 43
5.3.3.1 Data Compression Page 46
5.3.3.2 Device configuration page 48
5.3.3.3 Medium partition page(1) 51
5.3.3.4 Medium partition page(2-4) 55

5.3.3.5 Read-write error recovery page 56
5.4 Definitions specific to sequential access devices 59

6 Printer devices 61
6.1 Model for printer devices 61
6.2 Commands for printer devices 61
6.1.1 FORMAT command 62
6.1.2 PRINT 63
6.1.3 RECOVER BUFFERED DATA command 64
6.1.4 SLEW AND PRINT command 64
6.1.5 STOP PRINT command 65
6.1.6 SYNCHRONIZE BUFFER command 65
6.2 Parameters for printer devices 66
6.2.1 Diagnostic parameters 66
6.2.2 Log parameters 66
6.2.3 Mode parameters 67
6.2.3.1 Parallel printer interface page 68
6.2.3.2 Printer options page 69
6.2.3.3 Serial printer interface page 73

7 Communications devices 75
7.1 Communications device model 75
7.1.1 Implementation examples 76
7.1.1.1 Host-to-host communications 76
7.1.1.2 Host-to-device communications 76
7.1.1.3 Multiple role communications 77
7.2 Command descriptions for communications devices 77
7.2.1 GET MESSAGE(6) command 78
7.2.2 GET MESSAGE(10) command 79
7.2.3 GET MESSAGE(12) command 79
7.2.4 SEND MESSAGE(6) command 80
7.2.5 SEND MESSAGE(10) command 80
7.2.6 SEND MESSAGE(12) command 81
7.3 Parameters for communication devices 81
7.3.1 Diagnostic parameters 81
7.3.2 Log parameters 81
7.3.3 Mode parameters 82
7.4 Definitions specific to communications devices 82

8 Informative Annex 84
8.1 Density codes 84

Figures

Page
Figure 1 - SCSI-3 Document Roadmap....2
Figure 2 - Typical volume layout....10
Figure 3 - Typical medium track layout....10
Figure 4 - Serpentine recording example....11
Figure 5 - Parallel recording example....11
Figure 6 - Helical scan recording example....11
Figure 7 - Early-warning example....12
Figure 8 - Partitioning example - one partition per track group....12
Figure 9 - Partitioning example - one partition per two track groups....13
Figure 10 - Partitioning example - two partitions per track group....13
Figure 11 - SCSI printer model....61

Tables

Page
Table 1 - Commands for sequential-access devices 18
Table 2 - ERASE command 19
Table 3 - FORMAT MEDIUM command 20
Table 4 - Format field definition 21
Table 5 - LOAD UNLOAD command 21
Table 6 - LOCATE command 22
Table 7 - READ command 23
Table 8 - READ BLOCK LIMITS command 25
Table 9 - READ BLOCK LIMITS data 25
Table 10 - READ POSITION command 26
Table 11 - READ POSITION data format, short form 27
Table 12 - READ POSITION data format, long form 28
Table 13 - READ REVERSE command 29
Table 14 - RECOVER BUFFERED DATA command 30
Table 15 - REPORT DENSITY SUPPORT command 31
Table 16 - REPORT DENSITY SUPPORT header 32
Table 17 - DENSITY SUPPORT data block for each density 32
Table 18 - REWIND command 35
Table 19 - SPACE command 35
Table 20 - Code field definition 36
Table 21 - VERIFY command 38
Table 22 - WRITE command 39
Table 23 - WRITE FILEMARKS command 40
Table 24 - Diagnostic page codes 42
Table 25 - Log page codes 42
Table 26 - Parameter codes for sequential-access device page 43
Table 27 - Device-specific parameter 43
Table 28 - Buffered modes 44
Table 29 - Speed field definition 44
Table 30 - Sequential-access density codes 45
Table 31 - Mode page codes 45
Table 32 - Data compression page 46
Table 33 - Possible boundaries and resulting sense keys due to data compression 47
Table 34 - Compression algorithm identifiers 48
Table 35 - Device configuration page 49
Table 36 - EOD formats 50
Table 37 - Medium partition page(1) 51
Table 38 - PSUM 52
Table 39 - Medium format recognition 54
Table 40 - Medium partition page(2-4) 55
Table 41 - Read-write error recovery page 56
Table 42 - Commands for printer devices 62
Table 43 - FORMAT command 63
Table 44 - Format type 63
Table 45 - PRINT command 63
Table 46 - RECOVER BUFFERED DATA command 64
Table 47 - SLEW AND PRINT command 64
Table 48 - STOP PRINT command 65
Table 49 - SYNCHRONIZE BUFFER command 66
Table 50 - Diagnostic page codes 66
Table 51 - Log page codes 67
Table 52 - Printer device-specific parameter 67
Table 53 - Mode page codes 68

Table 54 - Parallel printer interface 68
Table 55 - Parity select 68
Table 56 - VFU control byte 69
Table 57 - Printer options 70
Table 58 - Font identification 70
Table 59 - Slew mode 71
Table 60 - Line slew 72
Table 61 - Form slew 72
Table 62 - Data termination option 73
Table 63 - Serial printer interface 73
Table 64 - Parity selection 74
Table 65 - Pacing protocol 74
Table 66 - Commands for communications devices 78
Table 67 - GET MESSAGE(6) command 78
Table 68 - GET MESSAGE(10) command 79
Table 69 - GET MESSAGE(12) command 79
Table 70 - SEND MESSAGE(6) command 80
Table 71 - SEND MESSAGE(10) command 80
Table 72 - SEND MESSAGE(12) command 81
Table 73 - Diagnostic page codes 81
Table 74 - Log page codes 82
Table 75 - Mode page codes 82
Table 76 - Historical sequential-access density codes 84

Notes

Page
NOTE 1 19
NOTE 2 21
NOTE 3 21
NOTE 4 24
NOTE 5 24
NOTE 6 24
NOTE 7 27
NOTE 8 28
NOTE 9 29
NOTE 10 30
NOTE 11 33
NOTE 12 33
NOTE 13 34
NOTE 14 34
NOTE 15 35
NOTE 16 36
NOTE 17 36
NOTE 18 38
NOTE 19 39
NOTE 20 40
NOTE 21 40
NOTE 22 40
NOTE 23 40
NOTE 24 41
NOTE 25 41
NOTE 26 42
NOTE 27 43
NOTE 28 47
NOTE 29 50
NOTE 30 51
NOTE 31 51
NOTE 32 52
NOTE 33 52
NOTE 34 52
NOTE 35 52
NOTE 36 53
NOTE 37 54
NOTE 38 54
NOTE 39 55
NOTE 40 55
NOTE 41 55
NOTE 42 56
NOTE 43 56
NOTE 44 68
NOTE 45 69
NOTE 46 69
NOTE 47 69
NOTE 48 76

Foreword

The Stream Device Command Set (SSC) standard specifies the external behavior of a device server that defines itself either a Stream-access device or a Printer device in the device type field of the INQUIRY command response data. Together, these device types are known as Sequential Devices. This standard applies only when the stream device is addressed as an independent logical unit. The SSC standard is specified independently of any service delivery subsystem. The SSC standard conforms to the minimum requirements specified in the SCSI-3 Architecture Model (SAM) standard.

Introduction

The Stream Device Command Set (SSC) standard specifies a protocol for command-level communications between an initiator and a device server which has identified itself as a stream device. No service delivery subsystem dependencies are included in this standard. The SSC standard originated from the command sets for Sequential-access, Printer, Processor, and Communications devices in the SCSI-2 (X3T9.2 Project 375D, rev 10l) standard.

The Stream Device Command Set (SSC) encompasses the following:

Clause 1 describes the scope.
Clause 2 lists the normative references.
Clause 3 provides a glossary common to the whole document.
Clause 4 provides descriptions and conventions.
Clause 5 specifies commands for sequential-access devices.
Clause 6 specifies commands for printer devices.
Clause 7 specifies commands for processor devices.
Clause 8 specifies commands for communications devices.

1 Scope

This standard defines the command interconnect between initiators and stream device device servers. The SSC standard assumes a service delivery subsystem exists which conforms to the SCSI-3 Architectural Model (SAM) standard.

Four device class command sets are included in this standard. These command sets originated from the SCSI-2 standard (X3.131-199x) chapters 10 (Sequential-access devices), 11 (Printer devices), 12 (processor devices), and 18 (Communications devices).

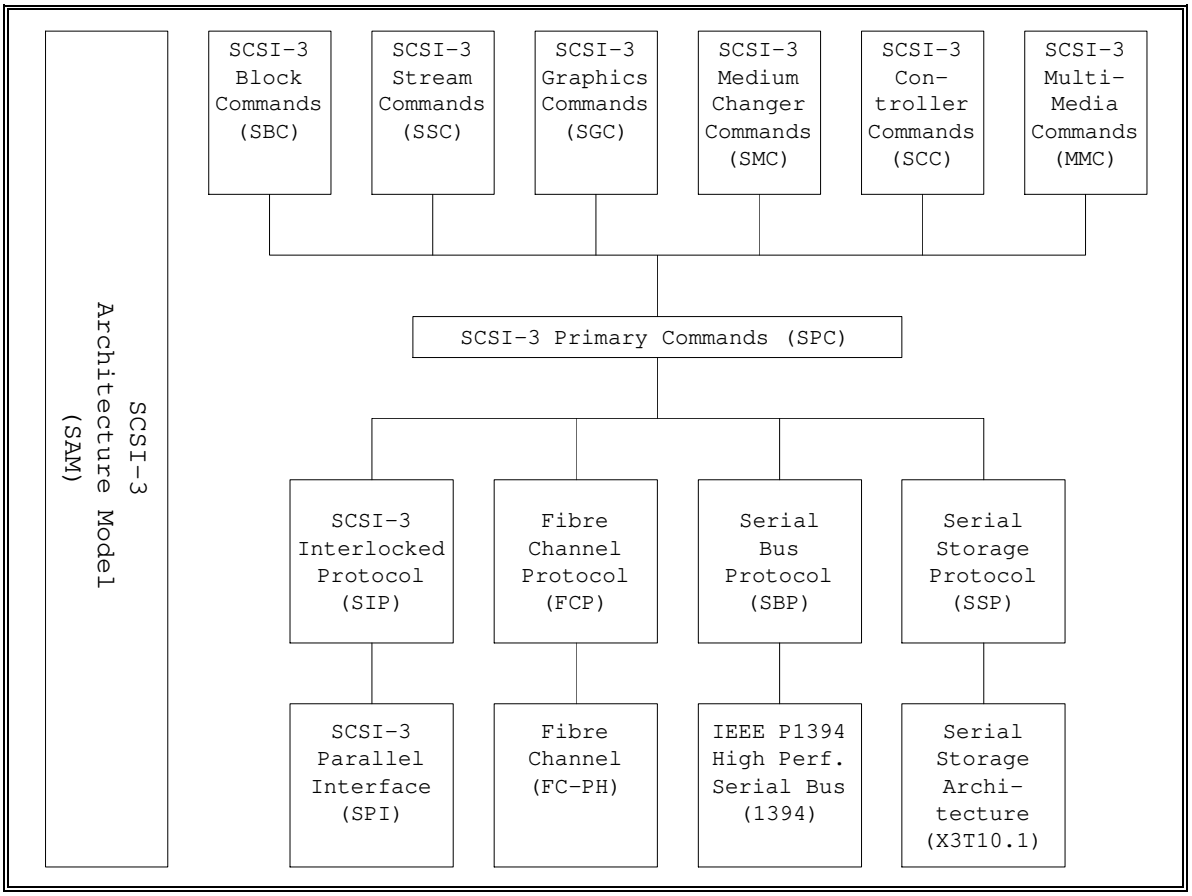

Figure 1 - SCSI-3 Document Roadmap

This roadmap is intended to show the general applicability of documents to one another.

The term SCSI is used whenever it is not necessary to distinguish between the versions of SCSI. The Small Computer System Interface - 2 standard (X3.131-1994) and the architecture that it describes are referred to herein as SCSI-2.

The term SCSI-3 refers collectively to the following documents that fall under the jurisdiction of X3T10:

FCP	X3T10/993D	SCSI-3 Fiber Channel Protocol
MMC	X3T10/1048D	SCSI-3 Multimedia Commands
SAM	X3T10/994D	SCSI-3 Architecture Model
SBC	X3T10/996D	SCSI-3 Block Commands
SBP	X3T10/992D	SCSI-3 Serial Bus Protocol
SCC	X3T10/1047D	SCSI-3 Controller Commands
SGC	X3T10/998D	SCSI-3 Graphic Commands
SIP	X3T10/856D	SCSI-3 Interlocked Protocol
SMC	X3T10/999D	SCSI-3 Medium Changer Commands
SPC	X3T10/995D	SCSI-3 Primary Commands
SPI	X3T10/855D	SCSI-3 Parallel Interface
SSC	X3T10/997D	SCSI-3 Stream Commands

Documents that are not listed above but are found in Figure 1 are not under the jurisdiction of X3T10.

2 Normative references

2.1 Corequisite Standards

The following standards contain provisions, which, through reference in this text, constitute provisions of this International 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 following lists of standards. Members of IEC and ISO maintain registers of currently valid International Standards. ANSI performs a similar function for American National Standards.

ANSI X3T10/994D, SCSI-3 Architecture Model (SAM).

ANSI X3T10/995D, SCSI-3 Primary Commands (SPC).

3 Definitions, symbols and abbreviations

This clause contains a glossary of special terms used in this standard. These terms apply to SSC and do not constitute a comprehensive glossary for SCSI-3.

3.1 Definitions

For the purposes of this standard, the following terms apply. These terms do not constitute a comprehensive glossary for SSC. Terms specific to a particular clause in this standard are defined in the appropriate clause.

3.1.1 **application client**: An object that is the source of SCSI commands. Further definition of an application client can be found in the SCSI-3 Architecture Model (SAM).

3.1.2 **byte**: Indicates an 8-bit construct.

3.1.3 **command**: A request describing a unit of work to be performed by a device server. A detailed definition of a command can be found in the SAM.

3.1.4 **command descriptor block (CDB)**: The structure used to communicate commands from an initiator to a target.

3.1.5 **device server**: An object within a logical unit that executes SCSI tasks according to the rules of task management. A detailed definition of a device server can be found in the SAM.

3.1.6 **field**: A group of one or more contiguous bits.

3.1.7 **initiator**: An SCSI device containing application clients that originate device service requests to be processed by a target SCSI device. A detailed definition of initiator can be found in the SAM.

3.1.8 **invalid**: An illegal (reserved) or unsupported field or code value.

3.1.9 **logical unit**: An externally addressable entity within a target that implements a SCSI device model. A detailed definition of a logical unit can be found in the SAM.

3.1.10 **mandatory**: The referenced item is required to claim compliance with this standard.

3.1.11 **one**: A true signal value or a true condition of a variable.

3.1.12 **optional**: The referenced item is not required to claim compliance with this standard. Implementation of an optional item must be as defined in this standard.

3.1.13 **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.

3.1.14 **peripheral device**: A physical peripheral device that can be attached to an SCSI device, which connects to the SCSI bus. The peripheral device and the SCSI device (peripheral controller) may be physically packaged together. Often there is a one-to-one mapping between peripheral devices and logical units, but this is not required. Examples of peripheral devices are: magnetic disks, printers, optical disks, and magnetic tapes.

3.1.15 **reserved**: Identifies bits, fields, and code values that are set aside for future standardization.

3.1.16 **SCSI device**: A device that is connected to a service delivery subsystem and supports a SCSI application protocol. A detailed definition of a SCSI device can be found in the SAM.

3.1.17 **target**: A SCSI device containing logical units that receive and execute commands from an initiator. A detailed definition of a target can be found in the SAM.

3.1.18 **vendor-specific (VS)**: Something (e.g. a bit, field, code value, etc.) that is not defined by this standard and may be used differently in various implementations.

3.1.19 **zero**: A false signal value or a false condition of a variable.

3.2 Symbols and abbreviations

BOM	Beginning-of-medium
BOP	Beginning-of-partition
EOM	End-of-medium
EOP	End-of-partition
EW	Early-warning
LSB	Least significant bit
LUN	Logical unit number (see clause 3.1.9)
MSB	Most significant bit
SAM	SCSI-3 Architecture Model (see clause 1)
SCSI	One of SCSI-2 or SCSI-3
SCSI-2	The architecture defined by the Small Computer System Interface -2 standard (ANSI X3.131:1994)
SCSI-3	The archicture defined by the family of standards described in clause 1
SPC	SCSI-3 Primary Commands (see clause 1)
SSC	SCSI-3 Stream Commands (this document, see clause 1)

This Page is Blank

4 General

4.1 Overview

The SCSI-3 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. Both have the common characteristic of primarily handling data in a sequential method, a stream. This does not limit the device's ability to position randomly within the data. However, the model for these devices implies that random-access of data may be impossible or at least inefficient.

A sequential-access device is not truly random-access (see the SPC standard for a comparison). Commands to read and write on a sequential-access device do not contain any absolute location. Instead, the device is positioned through other commands before the read or write. Writing to a sequential-access device does not merely replace the section, it may cause all data following that point to be invalidated. There may be restrictions on where the write can be initiated as well. Reading or writing data as a long string of data, as in a stream, tends to be the most efficient.

A printer device is like a sequential-access device since it also transfers data based on a current position, not an absolute location. However, a printer is different in that it is a data sink and is not expected to return the same data back during normal operation.

The physical model of both device types is very similar. For a sequential-access device, a recording media exists between two pools, the supply and take-up. The read/write mechanism can only access the media between the pools. As media is taken out of one, usually the supply, it passes by the read/write mechanism and into the take-up pool. Thus, transferring data as a stream is most efficient, since the media can traverse the read/write mechanism as a flow of data. To position to a given point requires moving the media until the appropriate position is found.

In the printer model, the read/write mechanism can only write and the media can only move from the supply. Positioning is limited to skipping forward in the media. Unlike a sequential-access device, data may be formatted on a printer device by font selection and positioning options. Therefore, a printer device class is not a strict subset of a sequential-access device class.

Both device classes have different levels of implementation. Depending on the desired capability of the logical unit, any number of the optional commands may be available. Restrictions on the use of the various commands may also depend on the device characteristics, independent on the command set implementation.

4.2 Conventions

Certain words and terms used in this International Standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 3 or in the text where they first appear. Names of signals, phases, messages, commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g. REQUEST SENSE). Lower case is used for words having the normal English meaning.

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 immediately followed by lower-case h (xxh) are hexadecimal values.

5 Sequential-access devices

5.1 Sequential-access device model

5.1.1 Physical elements

Sequential-access devices (called devices below) 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 direct-access devices.

Sequential-access devices are usually tape devices. The remainder of this description is from the point of view of a tape device; however, other implementations are not precluded.

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.

A complete unit composed of the recording medium and its physical carrier (e.g. reel, cartridge, cassette) is called a volume. 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 executing 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 can be executed. The logical unit is not ready when no volume is mounted or, from the initiator's perspective, whenever all medium access commands report CHECK CONDITION status and a NOT READY sense key. The logical unit is not ready during the transition between mounted and not mounted, or not mounted to mounted. Some devices may have a separate switch function which places the device in a not ready state even when a volume is mounted.

The Reserve and Release commands (see SPC) are mandatory for sequential-access devices. Extent reservations are not supported by this model.

The write enabled or write protected state determines when an initiator may write information on a volume. This attribute is usually controlled by the user of the volume through manual intervention (e.g. thumbwheel switch).

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.

As shown in figure 2, the entire physical length of medium is not usable for recording data. For most volumes, a length of the medium is reserved before the beginning-of-medium and after the end-of-medium position. This is done to provide sufficient tape wraps onto the reel hub(s) and to ensure that recording starts in an undamaged section of the medium.

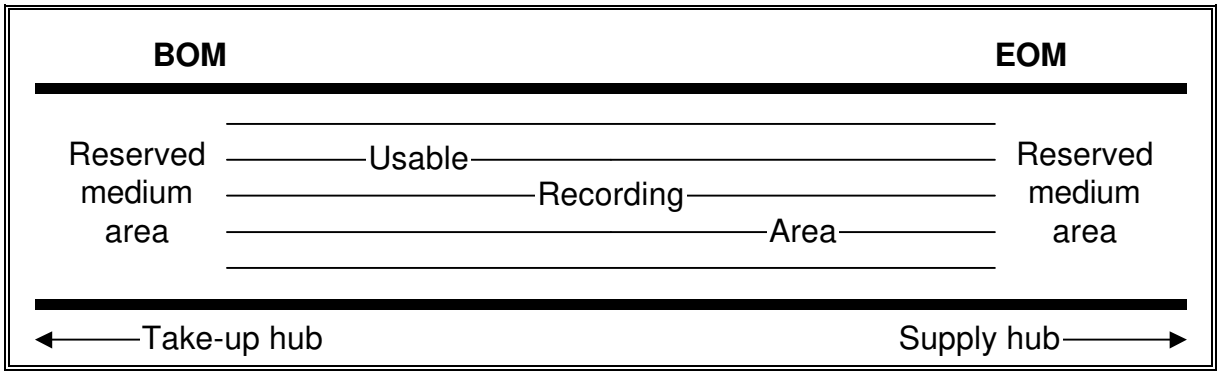

Figure 2 - Typical volume layout

5.1.2 Data storage characteristics

The position on the medium where a pattern of recorded signals may be written by one write component is called a track (see figure 3). A device may write or read from one or more tracks at a time, depending on the format.

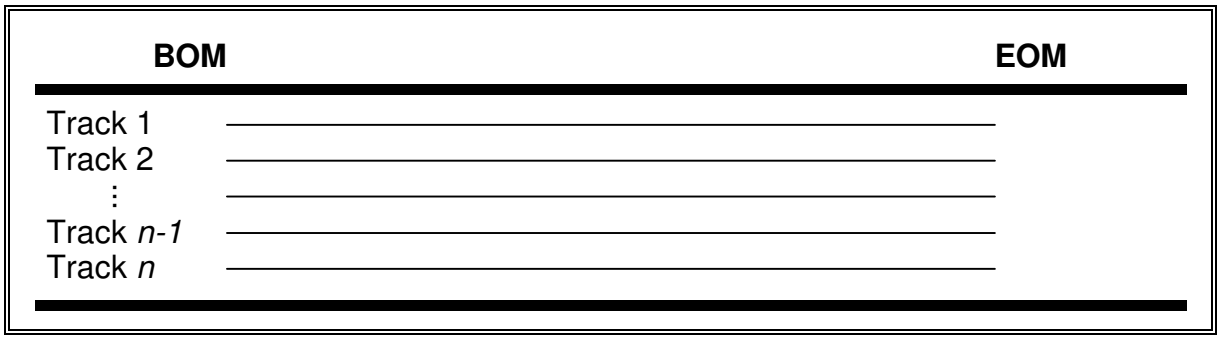

Figure 3 - Typical medium track layout

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

If not all tracks are recorded at the same time, and the device reverses direction when approaching end-of-medium and begins writing on remaining tracks, the recording method is called serpentine. For serpentine devices that record only one track at a time, each physical track represents one track group (see figure 4).

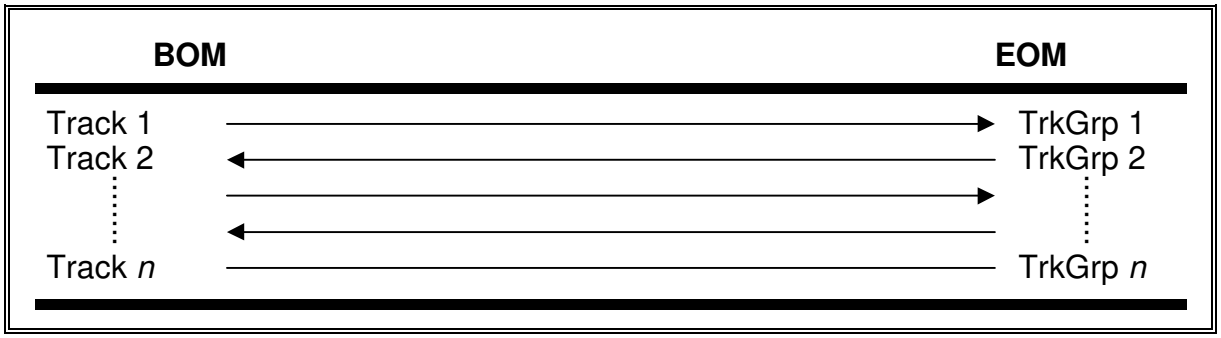

Figure 4 - Serpentine recording example

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 5).

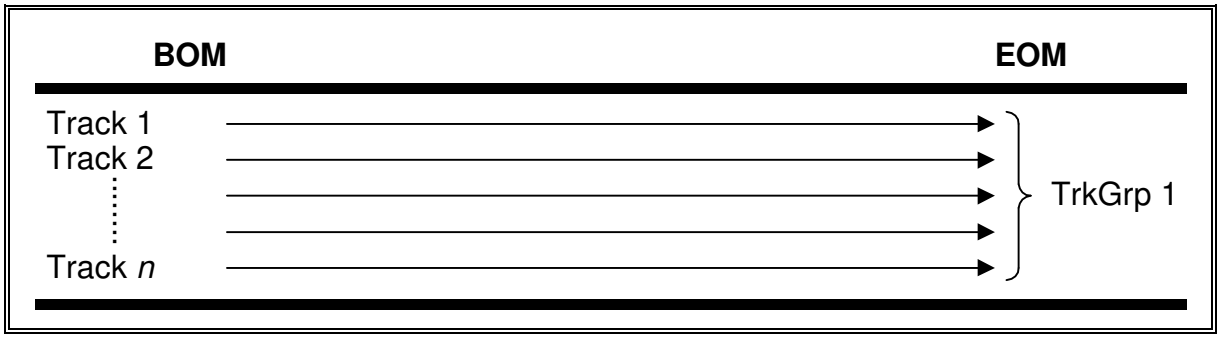

Figure 5 - Parallel recording example

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. This recording technique is known as helical scan (see figure 6).

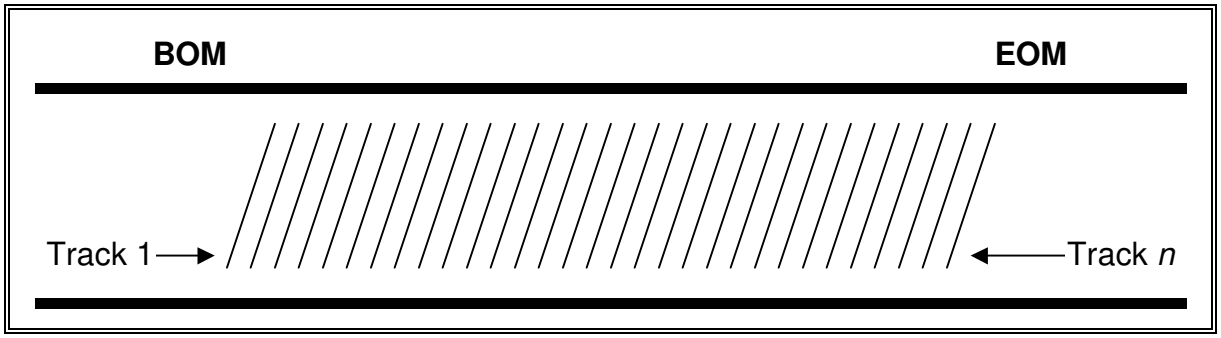

Figure 6 - Helical scan recording example

For most recording formats, an area at beginning-of-medium contains a format identification in the form of a tone burst or some other recognizable pattern. User data is not recorded in this area. The format identification is an attribute of a volume used for interchange purposes and is defined in applicable standards.

When writing, the initiator needs an indication that it is approaching the end of the permissible recording area. This position, called early-warning (EW), is typically reported to the initiator at a position early enough for the device to write any buffered data to the medium while still leaving enough room for

additional recorded labels or filemarks. Some American National Standards include physical requirements for a marker placed on the medium to be detected by the device as early-warning (see figure 7).

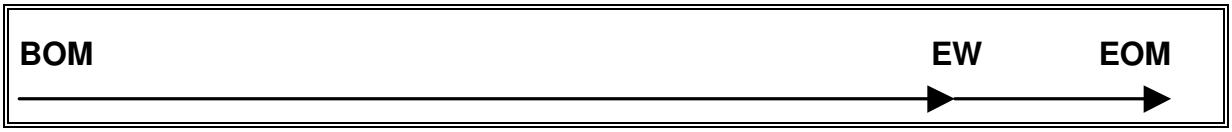

Figure 7 - Early-warning example

For devices which implement large data buffers, the early-warning position defined by a physical marker may be too close to the end of the recording region to permit emptying the data buffer(s). For these devices, a logical concept of early-warning is often used to signal the initiator at an appropriate location prior to the physical marker.

5.1.3 Partitions within a volume

Another attribute of a volume is called a partition. Partitions consist of one or more non-overlapped mini-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 which 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.

When a volume is mounted, it is logically positioned to 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 (BOP *x*).

Partitions on a volume do not need to be recorded in any defined order, nor do all partition numbers in a sequence need to be present on a volume. It is sufficient for a device to be able to locate a partition, given its code value, or 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 device-defined area (possibly unavailable to the initiator) or the information may be an intrinsic attribute of the device implementation.

Figure 8 shows a possible partition implementation for a four-track serpentine recording device, assuming that each track group defines a partition.

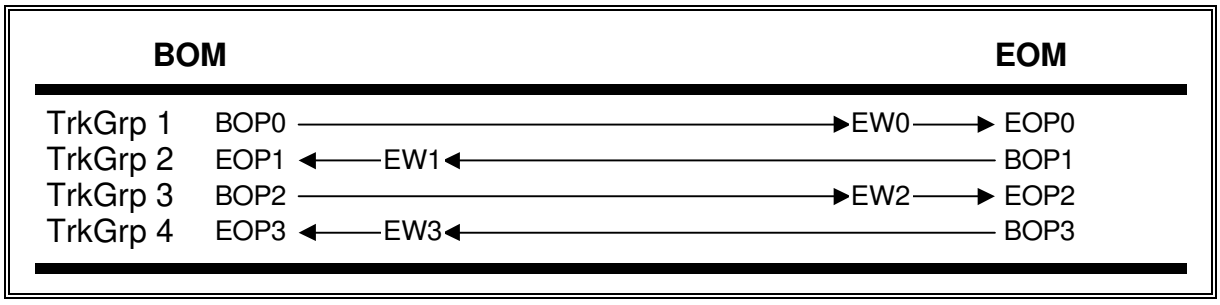

Figure 8 - Partitioning example - one partition per track group

Another possible partition implementation for this four-track serpentine recording device is shown in figure 9, using two track groups to define each partition.

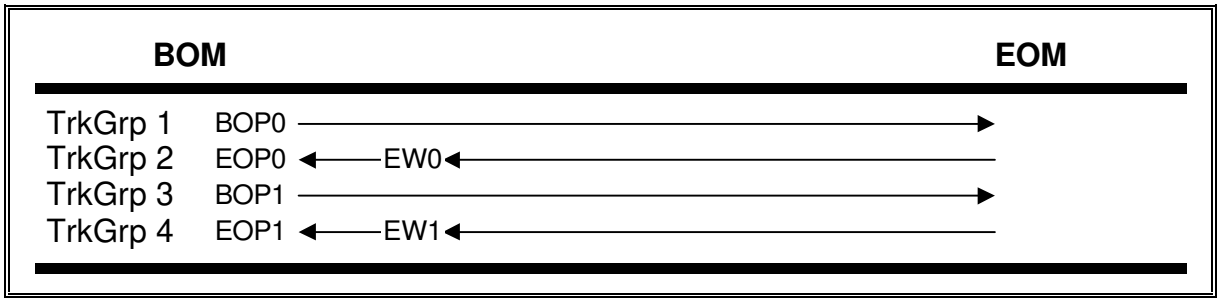

Figure 9 - Partitioning example - one partition per two track groups

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 recorded mark or some other device-recognizable attribute could be used to delineate the partitions. Figure 10 shows a possible two-partition implementation for a device with only one track group.

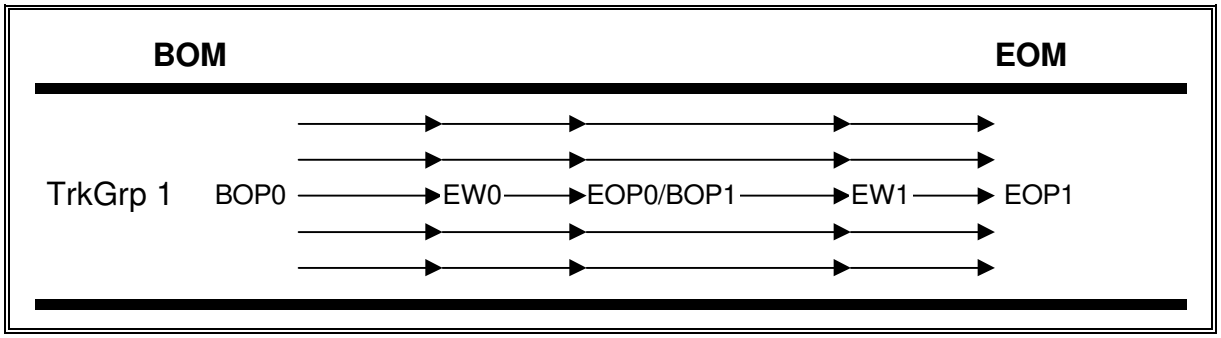

Figure 10 - Partitioning example - two partitions per track group

Three methods are defined in the MODE SENSE and MODE SELECT commands for managing partitions; each method is progressively more difficult to implement and manage:

a) device-defined fixed locations;
b) device-defined based on an initiator supplied number of partitions and a device specific allocation algorithm; and,
c) definition by name and capacity by an initiator.

5.1.4 Logical elements within a partition

The area between BOP *x* and EOP *x* on a typical recorded volume contains at least two types of initiator accessible elements, data blocks and tape marks. These elements are controlled and transferred between the initiator and the medium using READ, READ REVERSE, WRITE, and WRITE FILEMARKS commands.

A unit of data supplied or requested by an initiator is called a logical block. Logical blocks are stored according to the specifications of the format for the volume and may be recorded as one or more physical blocks on the medium. When the physical block and the logical block are not recorded in a one-to-one relationship, it is the responsibility of the device to perform all blocking, de-blocking, padding, stripping, splitting or rebuilding of the logical data block(s) sent by an application client.

Filemarks are special recorded elements containing no user data. The filemark format is defined in some American National Standards. Initiators traditionally use filemarks to separate user data from labels and logical groupings of data from each other. Since some format standards do not define an explicit end-of-data (EOD), host software has often used conventions with filemarks to represent an EOD

indication. At least one American National Standard specifically defines filemark use for this purpose. In some implementations, the device's EOD definition may be specified by the application client using the MODE SELECT command.

A setmark is another type of special recorded element containing no user data, providing a segmentation scheme hierarchically superior to filemarks. This level of segmentation is useful for some high capacity storage devices to provide concise addressing and fast positioning to specific sets of data within a partition. In some implementations, the detection and reporting of setmarks may be controlled by the application client using the MODE SELECT command.

Inter-block gaps, the gaps between blocks, filemarks, and setmarks, are introduced on the medium at the time a block or mark is written without explicit action by the initiator. Minimum and maximum lengths for inter-block gaps are defined in some American National Standards. In some devices, the length of inter-block gaps may be selected by the application client using the MODE SELECT command while in other devices the gaps are fixed and non-changeable.

In addition to blocks, filemarks, and setmarks, 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 block or mark. Minimum and maximum lengths for erase gaps are defined in some American National Standards while some devices may have no implementation of an erase gap.

After writing data from BOP *x*, the medium is considered to be a contiguous grouping of blocks, filemarks, setmarks, and gaps. Certain American National Standards define gap lengths which, if exceeded, are to be considered as having reached blank medium. 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 or variable length blocks. The concept of fixed or variable mode for writing and reading blocks only indicates 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. The length of a physical block may or may not be recorded as an exact byte count, depending on the format.

5.1.5 Data buffering

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

A device with a data buffer may be capable of operating in either a buffered mode or an unbuffered mode. A device with no data 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 a data buffer.

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

When issuing a buffered WRITE FILEMARKS command with the immediate bit set to one, GOOD status is returned as soon as the command is validated. A WRITE FILEMARKS command with the immediate bit set to zero causes any buffered blocks, filemarks, and setmarks to be written to the medium. Upon successful completion of this process, which is called a synchronize operation, no blocks, filemarks, or setmarks remain in the data buffer which have not been written to the medium. A synchronize operation has no effect on a data buffer which contains only read-ahead data or write data which has already been successfully written to the medium.

Should an unrecoverable write error occur 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). For some implementations, asynchronous event notification or extended contingent allegiance may be required. Refer to SPC for descriptions of asynchronous event notification and extended contingent allegiance protocol.

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

A device with read-ahead data blocks in the data buffer does not report an unrecovered read error until the data block in error is requested by an application client.

Prior to performing some commands, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium as stated in table 1. The WRITE BUFFER command shall ensure transfer of buffered data for modes 4 through 7 (firmware downloads). The MODE SELECT command shall ensure transfer of buffered data before the logical unit partitions the medium. The SEND DIAGNOSTICS command shall ensure transfer of buffered data before any diagnostic tests are initiated.

5.1.6 Recorded object descriptors (block identifiers)

Some recording formats specify that recorded objects (blocks, filemarks, and setmarks) have identifiers included in the recorded information to help determine write sequence and also to help detect device positioning errors. The identifier values are unique within a partition and may be unique within a volume.

The use of the term block identifier may imply some arithmetic sequence applied to the assignment of recorded objects. The block identifier assignment algorithm may be defined in an applicable format standard.

For some pre-formatted volumes, the identifiers are associated with physical blocks. In variable-length implementations, the identifier can be associated with a physical block when the logical block and the physical block have a one-to-one relationship on the medium.

Some formats may carry both physical and logical block identifiers recorded on the medium. When a logical block is split over more than one physical block, or multiple logical blocks are concatenated to form a physical block, the logical block identifier and the physical block identifier are not the same. Filemarks and setmarks may or may not have recorded identifiers, but if identifiers are used in the format, then each mark is assigned a value even if it is not explicitly recorded.

The READ POSITION and LOCATE commands use four-byte fields to hold these format dependent identifiers. For some implementations, this value may correspond to a real physical location; however, it is sufficient for the device to map the identifier to a value representing the unique recorded object. With this capability, the READ POSITION command may be used to report a device-defined block identifier and the initiator may use this value with a LOCATE command to position to the same location at some future time (provided the volume has not been rewritten in the interim).

Each logical element (data block, filemark, or setmark) has an unique block identifier on the medium. If supported, the end-of-data block identifier representing the position past the last logical element in a partition shall be unique for the medium.

5.1.7 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. This definition means 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 buffers have some physical position which 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 (BOP0). 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 indicates being positioned to a logical or physical extremity of a partition. A sequential-access device may be positioned at beginning-of-medium, at BOP *x*, at end-of-data (EOD), at EOP *x*, or at end-of-medium (EOM), since these are stable positions at extremities of a partition.

The concept of being before some position indicates that there is some element (data block, filemark, setmark, or other defined point) which may be encountered when moving toward EOP *x*, if the proper commands are issued. Being positioned before a particular data block means that if the device receives a valid READ command, the data 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 indicates that there is some element (data block, filemark, setmark, or other defined point) on the BOP *x* side of the current position which may be encountered if the proper commands are issued. When a READ command for a single data block has been successfully executed, the logical position is after the transferred data block.

5.1.8 Error reporting

If any of the following conditions occur during the execution of a command the device server shall return CHECK CONDITION status. The appropriate sense key and additional sense code should be set. The following list illustrates some error conditions and the applicable sense keys. The list does not provide an exhaustive enumeration of all conditions that may cause the CHECK CONDITION status.

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Unsupported option requested	ILLEGAL REQUEST
Target reset or medium change since last command from this initiator	UNIT ATTENTION
Self diagnostic failed	HARDWARE ERROR
Unrecovered read error	MEDIUM ERROR HARDWARE ERROR
Recovered read or write error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED command
Attempt a WRITE, READ, READ REVERSE, VERIFY, or RECOVER BUFFERED DATA command with the fixed bit set to zero and variable block mode is 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 execute an erase or write-type operation on write protected medium	DATA PROTECT
Deferred write error	MEDIUM ERROR VOLUME OVERFLOW HARDWARE ERROR

In the case of an unrecovered read or write error, if the read-write error recovery page (see 5.3.3.5) is implemented, the current values specify the logical unit error recovery criteria. If this page is not implemented, the error recovery is vendor-specific.

In the case of an unrecovered read error, if the fixed bit is one, the valid bit shall be set to one and the information field shall be set to the requested transfer length minus the actual number of blocks read (not including the unrecovered block). If the fixed bit is zero, the 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 block.

In the case of an unrecovered write error, if unbuffered mode is selected and the fixed bit is set to one, the valid bit shall be set to one and the information field shall be set to the requested transfer length minus the actual number of blocks written. If unbuffered mode is selected and the fixed bit is set to zero, the information field shall be set to the requested transfer length.

In the case of an unrecovered write error or a deferred write error, if buffered mode is selected and the fixed bit is one, the valid bit shall be set to one and the information field shall be set to the total number of blocks, filemarks, and setmarks not written (the number of blocks not transferred from the Data-Out buffer plus the number of blocks, filemarks, and setmarks remaining in the logical unit's buffer). If buffered mode is selected and the fixed bit is zero, the information field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of bytes not transferred from the Data-Out

buffer plus the number of bytes, filemarks, and setmarks remaining in the logical unit's buffer). In both cases, the value in the information field may exceed the transfer length.

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) or the first initiator responding to asynchronous event notification. If buffered mode 2h is selected, the error shall be reported to the initiator with unwritten data in the buffer.

5.2 Command descriptions for sequential-access devices

The commands for sequential-access devices shall be as shown in table 1. The Flush column indicates whether the command requires buffered data, filemarks, and setmarks to be transferred to the medium.

Table 1 - Commands for sequential-access devices

Command Name	Flush Required	Operation Code	Type	Subclause
CHANGE DEFINITION	Yes	40h	O	SPC
COMPARE	Yes	39h	O	SPC
COPY	Yes	18h	O	SPC
COPY AND VERIFY	Yes	3Ah	O	SPC
ERASE	Yes	19h	M	5.2.1
FORMAT	Yes	04h	O	5.2.2
INQUIRY	No	12h	M	SPC
LOAD UNLOAD	Yes	1Bh	O	5.2.3
LOCATE	Yes	2Bh	O	5.2.4
LOG SELECT	No	4Ch	O	SPC
LOG SENSE	No	4Dh	O	SPC
MODE SELECT (6)	Maybe	15h	M	SPC
MODE SELECT (10)	Maybe	55h	O	SPC
MODE SENSE (6)	No	1Ah	M	SPC
MODE SENSE (10)	No	5Ah	O	SPC
PREVENT ALLOW MEDIA REMOVAL	No	1Eh	O	SPC
READ	No	08h	M	5.2.5
READ BLOCK LIMITS	No	05h	M	5.2.6
READ BUFFER	No	3Ch	O	SPC
READ POSITION	No	34h	M	5.2.7
READ REVERSE	Yes	0Fh	O	5.2.8
RECEIVE DIAGNOSTIC RESULTS	No	1Ch	O	SPC
RECOVER BUFFERED DATA	No	14h	O	5.2.9
RELEASE UNIT	No	17h	M	SPC
REPORT DENSITY SUPPORT	No	44h	M	5.2.10
REQUEST SENSE	No	03h	M	SPC
RESERVE UNIT	No	16h	M	SPC
REWIND	Yes	01h	M	5.2.11
SEND DIAGNOSTIC	Maybe	1Dh	M	SPC
SPACE	Maybe	11h	M	5.2.12
TEST UNIT READY	No	00h	M	SPC
VERIFY	No	13h	O	5.2.13
WRITE	No	0Ah	M	5.2.14
WRITE BUFFER	Maybe	3Bh	O	SPC
WRITE FILEMARKS	Maybe	10h	M	5.2.15

Key: M = command implementation is mandatory.
O = command implementation is optional.
SPC = SCSI-3 Primary Command Set Standard.

The following command codes are vendor-specific: 02h, 06h, 07h, 09h, 0Ch, and 0Eh. All other command codes for sequential-access devices are reserved for future standardization.

5.2.1 ERASE command

The ERASE command (see table 2) causes part or all of the medium to be erased beginning at the current position on the logical unit. As used here, erased means either the medium shall be erased or a pattern shall be written on the medium that appears as a gap to the logical unit. Prior to performing the erase operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 2 - ERASE 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							

A reservation conflict shall occur when an ERASE command is received from an initiator other than the one holding a logical unit reservation.

An immediate (Immed) bit of zero indicates that the device server shall not return status until the erase operation has completed. An Immed bit of one indicates that the device server shall return status as soon as the command has been validated. If CHECK CONDITION status is returned for an ERASE command with an Immed bit of one, the erase operation shall not be performed.

A long bit of one indicates that 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 commands have completed execution and the command descriptor block of the ERASE command has been validated. The logical position following an ERASE command with a long bit of one is not specified by this standard.

NOTE 1 Some logical units may reject an ERASE command with the long bit set to one if the logical unit is not at beginning-of-partition.

A long bit of zero specifies an erase gap defined by the gap size field in the device configuration page (5.3.3.2). If the gap size is zero or the field is not supported, a device defined erase gap operation shall be performed. 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 initiator controlled error recovery or update in place applications.

If the logical unit encounters early-warning during an ERASE command, and any buffered data, filemarks, or setmarks remain to be written, the device server action shall be as defined for the early-warning condition of the WRITE command (5.2.14). If the long bit is zero, the erase operation shall terminate with CHECK CONDITION status and return sense data as defined for the WRITE command. Any buffered erases are not reported as part of the information field.

5.2.2 FORMAT MEDIUM command

The FORMAT MEDIUM command (see table 3) is used to prepare the medium for use by the logical unit. Prior to performing the format medium operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 3 - FORMAT MEDIUM command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (04h)							
1	Reserved						Verify	Immed
2	Reserved				Format			
2	(MSB) Transfer							
4	Length							(LSB)
5	Control							

The FORMAT MEDIUM command is an optional command for sequential access devices.

A reservation conflict shall occur when a FORMAT MEDIUM command is received from an initiator other than the one holding a logical unit reservation.

The FORMAT MEDIUM command shall be accepted only when the medium is at Beginning of Tape (BOT) or 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 with the additional sense code set to POSITION PAST BEGINNING OF MEDIUM.

A valid FORMAT MEDIUM command shall cause all data on the entire physical volume to be lost.

At the successful completion of a FORMAT MEDIUM command, the medium shall be positioned at BOT or BOP 0. Any prior data on the volume shall become inaccessible.

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 which case RESERVATION CONFLICT status shall be returned.
b) In response to the REQUEST SENSE command, assuming no error has occurred, the device server shall return a sense key of NOT READY with the additional sense code set to LOGICAL UNIT NOT READY -- FORMAT IN PROGRESS, with the sense key specific bytes set for process indication (as described in SPC).

An immediate (Immed) bit of zero indicates that the device server shall not return status until the FORMAT MEDIUM command has completed. An Immed bit of one indicates that 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 indicates that the logical unit shall verify that the format was successfully 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 with the additional sense code set to MEDIUM FORMAT CORRUPTED. If the verify field is zero, the logical unit shall not perform the verify check.

The format field is defined in table 4.

Table 4 - Format field definition

Value	Description	Support
0h	Use default format	Mandatory
1h – 7h	Reserved	
8h – Fh	Vendor-specific	

If the format field is 0h, the logical unit shall determine which format method to use.

The transfer length specifies the length in bytes of format information that shall be located in the Data-Out buffer. A transfer length of zero indicates that no format information shall be transferred. This condition shall not be considered an error. If the format field is 0h, the transfer length shall be zero.

NOTE 2 A logical unit may implement different format methods by any legal combination of format field values and format information data. This standard does not specify the format of the format information data.

5.2.3 LOAD UNLOAD command

The LOAD UNLOAD command (see table 5) 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 or unload operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 5 - 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					EOT	Reten	Load
5	Control							

A reservation conflict shall occur when a LOAD UNLOAD command is received from an initiator other than the one holding a logical unit reservation.

An immediate (Immed) bit of zero indicates that the device server shall not return status until the load or unload operation has completed. An Immed bit of one indicates that the device server shall return status as soon as all buffered commands have completed execution 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 3 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 ensure that all buffered data, filemarks, or setmarks have been transferred to the medium prior to issuing a LOAD UNLOAD command with an Immed bit of one.

An end-of-tape (EOT) bit of one indicates that an unload operation (load bit set to zero) shall position the medium at end-of-medium for removal from the device. An EOT bit of zero indicates that 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 in the sense data.

A retension (ReTen) bit of one indicates that the medium on the logical unit shall have the correct tension applied. Implementation of the retension function is device specific.

If the load bit is set to one, the medium in the logical unit shall be loaded and positioned to the beginning-of-partition zero. If the load bit is zero, 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.

When operating in buffered mode 1h or 2h (see 5.3.3), the logical unit shall discard any unwritten buffered data after the LOAD UNLOAD command is validated if the previous command was terminated with CHECK CONDITION status and the device is unable to continue successfully writing.

5.2.4 LOCATE command

LOCATE command (see table 6) causes the logical unit to position the logical unit to the specified block address in a specified partition. Upon completion, the logical position shall be before the specified location. Prior to performing the locate operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 6 - LOCATE 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	Block							
5	Address							
6								(LSB)
7	Reserved							
8	Partition							
9	Control							

A reservation conflict shall occur when a LOCATE command is received from an initiator other than the one holding a logical unit reservation.

A block address type (BT) bit of one indicates the value in the block address field shall be interpreted as a device-specific value. A BT bit of zero indicates the value in the block address field shall be interpreted as an SCSI logical block address.

A change partition (CP) bit of one indicates that a change to the partition specified in the partition field is to occur prior to positioning to the block specified in the block address field. A CP bit of zero indicates no partition change is to be made and the partition field is to be ignored.

An immediate (Immed) bit of zero indicates that the device server shall not return status until the locate operation has completed. An Immed bit of one indicates that the device server shall return status as soon as all buffered commands have completed execution and the command descriptor block of the LOCATE command has been validated. If CHECK CONDITION status is returned for a LOCATE command with an Immed bit of one, the locate operation shall not be performed.

The block address field (see 5.1.6) specifies the block address to which the logical unit shall position the medium based on the current setting of the BT bit. An otherwise valid LOCATE 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 to a position past end-of-data shall return CHECK CONDITION status and the sense key shall be set to BLANK CHECK. Additionally, the EOM bit shall be set to one if end-of-data is located at or after early-warning.

If the end-of-partition is encountered while spacing forward over blocks, filemarks, or setmarks, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, and the EOM bit shall be set to one.

The partition field specifies which partition to select if the CP bit is one. Refer to the sequential-access device model (see 5.1.3) and the medium partition pages (see 5.3.3.3 and 5.3.3.4) for additional information about partitioning.

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

5.2.5 READ command

The READ command (see table 7) requests that the device server transfer one or more block(s) of data to the application client beginning with the next block on the logical unit.

Table 7 - READ 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							

A reservation conflict shall occur when a READ command is received from an initiator other than the one holding a logical unit reservation.

The fixed bit specifies whether fixed-length or variable-length blocks are to be transferred. Refer to the READ BLOCK LIMITS command (see 5.2.6) for additional information about fixed and variable block mode.

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). 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 data.

A successful READ 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 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 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).

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 4 Since the residue 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 data 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 block is read, CHECK CONDITION status shall be returned and the ILI and valid bits shall be set to one in the sense data. Upon termination, the logical position shall be after the incorrect length 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 blocks read (not including the incorrect length block). If the fixed bit is zero, the information field shall be set to the requested transfer length minus the actual block length. Logical units that do not support negative values shall set the information field to zero if the overlength condition exists.

NOTE 5 In the above case with the fixed bit of one, only the position of the incorrect-length logical block can be determined from the sense data. The actual length of the incorrect 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 indicates that no data shall be transferred. This condition shall not be considered an error and the logical position shall not be changed.

If the device server encounters a filemark during a READ 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. 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 blocks read (not including the filemark). If the fixed bit is zero, the information field shall be set to the requested transfer length.

If the device server encounters a setmark during a READ command and the RSmk bit is set to one in the device configuration page (see 5.3.3.2), 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 SETMARK DETECTED. Upon termination, the logical position shall be after the setmark (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 blocks read (not including the setmark). If the fixed bit is zero, the information field shall be set to the requested transfer length. The device server shall not return CHECK CONDITION when a setmark is encountered if the RSmk bit is set to zero or if this option is not supported.

If the device server encounters early-warning during a READ command and the REW bit is set to one in the device configuration page (see 5.3.3.2), CHECK CONDITION status shall be returned upon completion of the current block. The sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. The EOM and valid bits shall be set to one in the sense data. Upon termination, the logical position shall be after the last block transferred (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 blocks read. If the fixed bit is zero, the information field shall be set to the requested transfer length minus the actual block length. The device server shall not return CHECK CONDITION status when early-warning is encountered if the REW bit is zero or if the REW option is not supported.

NOTE 6 A REW bit of one is not recommended for most system applications since read data may be present after early-warning.

If the device server encounters end-of-data during a READ 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 after the last recorded 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 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 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 blocks read. If the fixed bit is zero, the information field shall be set to the requested transfer length.

5.2.6 READ BLOCK LIMITS command

The READ BLOCK LIMITS command (see table 8) requests that the logical unit's block length limits capability be returned. The READ BLOCK LIMITS data shown in table 9 shall be returned.

Table 8 - 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							

A reservation conflict shall occur when a READ BLOCK LIMITS command is received from an initiator other than the one holding a logical unit reservation.

Table 9 - READ BLOCK LIMITS data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	(MSB)	Maximum block						
2		length limit						
3								(LSB)
4	(MSB)	Minimum block						
5		length limit						(LSB)

If the maximum block length limit value equals the minimum block length limit value, the logical unit supports the transfer of data in the fixed-block mode only, with the block length equal to the given value. In this case the fixed bit shall be set to one in the WRITE and READ commands.

If the maximum block length limit value is not equal to the minimum block length limit value, the logical unit supports the transfer of data in either fixed-block or variable-block modes, with the block length constrained between the given limits in either 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.

5.2.7 READ POSITION command

The READ POSITION command (see table 10) reports the current position of the logical unit and any data blocks in the buffer. No medium movement shall occur as a result of the command.

Table 10 - READ POSITION command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (34h)							
1	Reserved					TCLP	LONG	BT
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Control							

A reservation conflict shall occur when a READ POSITION command is received from an initiator other than the one holding a logical unit reservation.

A Total Current Logical Position (TCLP) bit of one indicates the device server shall return data specifying the partition, file, and set number with the current logical position. A TCLP bit of zero indicates the device server shall return data specifying the first and last block location with the number of bytes and blocks in the buffer. Support of TCLP is optional. If the device server does not implement Total Current Logical Position and the TCLP bit is set to one, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

A Long Format (LONG) bit of one indicates the device server shall return 32 bytes of data. A Long Format bit of zero indicates the device server shall return 20 bytes of data.

The Long Format (LONG) bit and the Total Current Logical Position (TCLP) bit shall be equal. If the Long Format and Total Current Logical Position bits are not equal, or if both the Long Format bit and the Block Address Type (BT) bits are one, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The Block Address Type (BT) controls the content of the short format data. A Block Address Type bit of one requests the device server to return the first block location, the last block location, and block number fields as device-specific values. A BT bit of zero requests the device server to return the first block location, the last block location, and number of blocks as an SCSI logical block address (relative to a partition).

The following READ POSITION data shall be returned if TCLP and LONG are set to 0.

Table 11 - READ POSITION data format, short form

Bit / Byte	7	6	5	4	3	2	1	0
0	BOP	EOP	BCU	BYCU	Rsvd	BPU	PERR	Rsvd
1	Partition Number							
2	Reserved							
3	Reserved							
4 : 7	(MSB) First block location (LSB)							
8 : 11	(MSB) Last block location (LSB)							
12	Reserved							
13 : 15	(MSB) Number of blocks in buffer (LSB)							
16 : 19	(MSB) Number of bytes in buffer (LSB)							

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

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

NOTE 7 The BOP and EOP indications are not necessarily a result of a physical tape marker (e.g. reflective marker).

A block count unknown (BCU) bit of one indicates that the Number of blocks in buffer field does not represent the actual number of blocks in the buffer. A BCU bit of zero indicates that the Number of blocks in buffer field is valid.

A byte count unknown (BYCU) bit of one indicates that the Number of bytes in buffer field does not represent the actual number of bytes in the buffer. A BYCU bit of zero indicates that the Number of bytes in buffer field is valid.

A block position unknown (BPU) bit of one indicates that the first and last block locations are not known or cannot be obtained. A BPU bit of zero indicates that the first and last block location fields contain valid position information. While the logical unit is processing an immediate command, READ POSITION shall set BPU to one in the returned data.

If the logical unit is not capable of reporting block locations, the BPU bit shall be set to one in the returned data. Support for this option of the READ POSITION command is indicated by a BIS bit set to one in the device configuration page (see 5.3.3.2).

A Position Error (PERR) bit of one indicates that the logical unit cannot report the correct position due to an overflow of any of the returned position data. A Position Error of zero indicates that 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, this field shall be set to zero.

The first block location field indicates the block address associated with the current logical position. The value shall indicate the block address of the next data block to be transferred between an application client and the device server if a READ or WRITE command is issued.

The last block location field indicates the block address (see 5.1.6) associated with the next block to be transferred from the buffer to the medium. The value shall indicate the block address of the next data block to be transferred between the buffer and the medium. If the buffer does not contain a whole block of data or is empty, the value reported for the last block location shall be equal to the value reported for the first block location.

NOTE 8 The information provided by the first and last block location fields can be used in conjunction with the LOCATE command to position the medium at the appropriate logical block on another device in the case of unrecoverable errors on the first device.

The number of blocks in buffer field indicates the number of data blocks in the logical unit's buffer that have not been written to the medium.

The number of bytes in buffer field indicates the total number of data bytes in the logical unit's buffer that have not been written to the medium.

The following READ POSITION data shall be returned if TCLP and LONG are set to 1.

Table 12 - READ POSITION data format, long form

Bit / Byte	7	6	5	4	3	2	1	0
0	BOP	EOP	Reserved		MPU	BPU	Reserved	
1	Reserved							
2	Reserved							
3	Reserved							
4 : 7	(MSB) Partition number (LSB)							
8 : 15	(MSB) Block number (LSB)							
16 : 23	(MSB) File number (LSB)							
24 : 31	(MSB) Set number (LSB)							

The BOP, EOP, BPU, and Partition Number fields are as defined in the READ POSITION data returned when TCLP is set to 0.

A Mark Position Unknown (MPU) bit of one indicates the File Number and Set Number are unknown. A MPU bit of zero indicates the File Number and Set Number fields contain valid position information. While the device server is processing an immediate command, READ POSITION shall set MPU to one in the returned data.

NOTE 9 If the logical unit is assuming a position due to unrecoverable read errors, the MPU and BPU fields should be set to zero where possible. The MPU and BPU bits shall only be set if the logical unit cannot assume or does not know the current mark or block, respectively.

The Partition Number shall report the partition number as specified by the short-format Read Position data.

The Block Number shall report the number of logical blocks between beginning-of-partition and the current logical position. Setmarks and filemarks count as one logical block each.

The File Number shall report the number of filemarks between beginning-of-partition and the current logical position.

The Set Number shall report the number of setmarks between beginning-of-partition and the current logical position.

5.2.8 READ REVERSE command

The READ REVERSE command (see table 13) requests that the device server transfer one or more block(s) of data to the application client beginning at the current position on the logical unit. Prior to performing the read operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 13 - READ REVERSE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (0Fh)							
1	Reserved						SILI	Fixed
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a READ REVERSE command is received from an initiator other than the one holding a logical unit reservation.

The execution of this command is similar to the READ command except that medium motion is in the reverse direction. All block(s), and the byte(s) within the block(s), are transferred in the reverse order. The order of bits within each byte shall not be changed. Upon completion of a READ REVERSE command, the logical position shall be before the last block transferred (beginning-of-partition side).

Refer to the READ command (see 5.2.5) for a description of the fixed bit, the SILI bit, the transfer length field, and any associated error conditions.

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

If the device server encounters beginning-of-partition during a READ REVERSE 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 blocks transferred. If the fixed bit is zero, the information field shall be set to the requested transfer length.

5.2.9 RECOVER BUFFERED DATA command

The RECOVER BUFFERED DATA command (see table 14) is used to recover data that has been transferred to the logical unit's buffer but has not been successfully written to the medium. It is normally used to recover from error or exception conditions that 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 14 - 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							

A reservation conflict shall occur when a RECOVER BUFFERED DATA command is received from an initiator other than the one holding a logical unit reservation.

The execution of this command is similar to the READ command except that the data is transferred from the logical unit's buffer instead of the medium. The order in which block(s) are transferred is defined by the RBO bit in the device configuration page (see 5.3.3.2). If the RBO bit is not implemented, block(s) are transferred in the same order they would have been transferred to the medium.

Refer to the READ command (see 5.2.5) for a description of the fixed bit, the SILI bit, the transfer length field, and any associated error conditions.

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 10 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 blocks transferred (not including the filemark).

If the fixed bit is zero, the information field shall be set to the requested transfer length.

If a buffered setmark is encountered during a RECOVER BUFFERED DATA command and the RSmk bit is set to one in the device configuration page (see 5.3.3.2), 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 SETMARK DETECTED. Upon termination, the logical position shall be after the setmark. If the fixed bit is one, the information field shall be set to the requested transfer length minus the actual number of blocks transferred (not including the setmark). If the fixed bit is zero, the information field shall be set to the requested transfer length. The device server shall not return CHECK CONDITION when a setmark is encountered if the RSmk bit is zero or if this option is not supported.

If an attempt is made to recover more logical blocks of data than are contained in the logical unit's buffer, 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. If the fixed bit is one, the information field shall be set to the requested transfer length minus the actual number of blocks transferred. If the fixed bit is zero, the information field shall be set to the requested transfer length.

5.2.10 REPORT DENSITY SUPPORT Command

The REPORT DENSITY SUPPORT command (see table 15) requests that information regarding the supported densities for the logical unit be sent to the application client. Most of the information reported by the REPORT DENSITY SUPPORT command is static. Support of this command is mandatory for sequential access devices. All supported densities shall be supported for reading.

Table 15 - REPORT DENSITY SUPPORT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation Code (44h)							
1	Reserved							Media
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Allocation							
8	Length							(LSB)
9	Control							

A reservation conflict shall occur when a REPORT DENSITY SUPPORT command is received from an initiator other than the one holding a logical unit reservation.

A Media bit of zero indicates that the device server shall return density support data blocks for densities supported by the logical unit for any supported media. The data returned by the device server shall be static if the Media bit is zero. A Media bit of one indicates that the device server shall return density support data blocks for densities supported by the mounted medium. If the Media bit is one and the logical unit is not in the ready state, CHECK CONDITION status shall be returned. The sense key shall be set to NOT READY.

The Allocation Length field specifies the maximum number of bytes that the device server may return.

The REPORT DENSITY SUPPORT command returns the REPORT DENSITY SUPPORT header (see table 16) followed by one or more DENSITY SUPPORT data blocks (see table 17). The Density support data blocks shall follow the REPORT DENSITY SUPPORT header. The Density Support data blocks shall be in numerical ascending order of the primary density code value for each block.

Table 16 - REPORT DENSITY SUPPORT header

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB) Available density							
1	support length							(LSB)
2	Reserved							
3	Reserved							
4 n	Density support data blocks for supported densities							

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. This field shall be equal to 2 more than an integer multiple of 52 (52 is the size of a Density support data block).

Table 17 - DENSITY SUPPORT data block for each density

Bit / Byte	7	6	5	4	3	2	1	0
0	Primary Density Code							
1	Secondary Density Code							
2	WrtOK	Dup	Deflt	Reserved				
3	Reserved							
4	Reserved							
5 : 7	(MSB) Bits per mm							(LSB)
8 9	(MSB) Media Width							(LSB)
10 11	(MSB) Tracks							(LSB)
12 : 15	(MSB) Capacity							(LSB)
16 : 23	Assigning organization							
24 : 31	Density Name							
32 : 51	Description							

Density support data blocks 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 other fields except for the Assigning organization, density name, and description shall be identical. Density support data blocks with the same primary density code value should be ordered from most to least preferred Assigning organization, Density Name, and Description.

The density support data block may represent a particular format in addition to giving physical density information. The information in a density support data block provides an application client with a detailed review of the recording technologies supported by a logical unit. By supplying the a density code value returned in a density support block in a MODE SELECT command, an application client can select 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 block. The device server shall accept a Mode Select command containing this value, for appropriate media. The value 07Fh 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. All density codes values returned by the Report Density Support command shall support read operations based on the setting of the Media bit in the CDB. When density information matches one of the entries in table 76 (annex 8.1), the Primary Density Code value should match the density code assigned in the table.

NOTE 11 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..

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. The Mode Select command 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 shall indicate that logical unit support for this density does not include writing to the media. A WrtOK bit of one shall indicate that the logical unit is capable of writing this density to either the currently mounted medium (Media bit in CDB set to one) or for some media (Media bit in CDB set to zero).

A Dup bit of zero shall indicate that this primary density code has exactly one density support data block. A Dup bit of one shall indicate that this primary density code is specified in more than one density support data block.

A Deflt bit of zero shall indicate that this density is not the default density of the drive. A Deflt bit of one shall indicate that this density is the default density of the logical unit. If either the Primary or Secondary density code 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 density code of 00h as equivalent to the Primary and Secondary density codes.

NOTE 12 The default density of the logical unit may vary depending on the media installed. Multiple codes canreturn a deflt bit of one when the media bit is zero since more than one default may be possible.

The Bits per mm field indicates 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 indicates that 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 indicates 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 indicates that the width of the medium does not apply to this logical unit.

The Tracks field indicates the number of data tracks supported on the medium by this density. The Tracks 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 which are neither parallel nor serpentine, the tracks field indicates the maximum number of data tracks that can be read or recorded simultaneously.

If the Media bit of the CDB is set to 0, the Capacity field shall indicate the approximate capacity of the longest supported medium assuming recording in this density with one partition. If the Media bit of the CDB is set to 1, the Capacity field should indicate the approximate capacity of the current medium, assuming recording in this density with one partition. If the approximate capacity of the current medium cannot be determined for the mounted medium, the longest supported medium capacity shall be used. 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 organization responsible for the specifications defining the values in this density support data block. 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 in the SPC. The use of a specific vendor identification, other than the one associated with the device is allowed. Thus, if vendor ABC defines a density and format, another vendor may use ABC in the Assigning organization field. If exactly the same density and format construction later becomes known by another name, both ABC and the new assigning organization may be used for the density code. This is one condition that may result in multiple density support data blocks for a single density code value.

NOTE 13 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. In the absence of any formal registration procedure, X3T10 maintains a list of vendors and assigning organizations identification codes in use. Vendors are requested to voluntarily submit their identification codes to X3T10 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. 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 name fields. Assigning organizations shall be responsible for preventing duplicate usage one density name for multiple different densities and/or formats.

NOTE 14 It is suggested that any document which 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.

5.2.11 REWIND command

The REWIND command (see table 18) causes the logical unit to position to the beginning-of-partition in the current partition. Prior to performing the rewind operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 18 - 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							

A reservation conflict shall occur when a REWIND command is received from an initiator other than the one holding a logical unit reservation.

An immediate (Immed) bit of zero indicates that the device server shall not return status until the rewind operation has completed. An Immed bit of one indicates that the device server shall return status as soon as all buffered commands have completed execution 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 15 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 ensure that all buffered data, filemarks, or setmarks have been transferred to the medium before issuing a REWIND command with an Immed bit of one.

When operating in buffered mode 1h or 2h (see 5.3.3), the logical unit shall discard any unwritten buffered data after the REWIND command is validated if the previous command was terminated with CHECK CONDITION status and the device is unable to continue successfully writing.

5.2.12 SPACE command

The SPACE command (see table 19) provides a variety of positioning functions that are determined by the code and count. Both forward and reverse positioning are provided, although some logical units may only support a subset of this command. If an application client requests an unsupported function, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUE Prior to performing the space operation, except as stated in the description of the count field, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.ST.

Table 19 - SPACE 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							

A reservation conflict shall occur when a SPACE command is received from an initiator other than the one holding a logical unit reservation.

The code field is defined in table 20.

Table 20 - Code field definition

Code	Description	Support
000b	Blocks	Mandatory
001b	Filemarks	Mandatory
010b	Sequential filemarks	Optional
011b	End-of-data	Optional
100b	Setmarks	Optional
101b	Sequential setmarks	Optional
110b-111b	Reserved	

When spacing over blocks, filemarks, or setmarks, the count field specifies the number of blocks, filemarks, or setmarks to be spaced over in the current partition. A positive value *N* in the count field shall cause forward positioning (toward end-of-partition) over *N* blocks, filemarks, or setmarks ending on the end-of-partition side of the last block, filemark, or setmark. A zero value in the count field shall cause no change of logical position. A negative value *-N* (two's complement notation) in the count field shall cause reverse positioning (toward beginning-of-partition) over *N* blocks, filemarks, or setmarks ending on the beginning-of-partition side of the last block, filemark, or setmark. Support of spacing in the reverse direction is optional. When the count field is zero and spacing over blocks, filemarks or setmarks, buffered data, filemarks, and setmarks may be transferred to the medium.

If a filemark is encountered while spacing over blocks, the command shall be terminated. 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. CHECK CONDITION status shall be returned to the application client, the sense key shall be set to NO SENSE, and the filemark and valid bits shall be set to one in the sense data. The information field shall be set to the requested count minus the actual number of blocks spaced over (not including the filemark).

NOTE 16 A CHECK CONDITION caused by early termination of any SPACE command never results in a negative information field value.

If a setmark is encountered while spacing over blocks or filemarks and the RSmk bit is set to one in the device configuration page (see 5.3.3.2), the command shall be terminated, CHECK CONDITION status shall be returned to the application client, 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 SETMARK DETECTED. The information field shall be set to the requested count minus the actual number of blocks or filemarks spaced over (not including the setmark). The logical position shall be on the end-of-partition side of the setmark if movement was in the forward direction and on the beginning-of-partition side of the setmark if movement was in the reverse direction. The device server shall not return CHECK CONDITION status when a setmark is encountered if the RSmk bit is set to zero or if this option is not supported.

If early-warning is encountered while spacing over blocks, filemarks, or setmarks and the REW bit is set to one in the device configuration page (see 5.3.3.2), 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 information field shall be set to the requested count minus the actual number of blocks, filemarks, or setmarks 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 17 Setting the REW bit is not recommended for most system applications since data may be present after early-warning.

If end-of-data is encountered while spacing over blocks, filemarks, or setmarks, 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. Additionally, the 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 blocks, filemarks, or setmarks spaced over as defined by the code value.

If the end-of-partition is encountered while spacing forward over blocks, filemarks, or setmarks, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, the EOM bit shall be set to one, and the valid bit shall be set to one. The information field shall be set to the requested count minus the actual number of blocks, filemarks, or setmarks spaced over as defined by the code value.

If beginning-of-partition is encountered while spacing over blocks, filemarks, or setmarks in the reverse direction, the device server shall return CHECK CONDITION status and shall set the sense key to NO SENSE. The EOM and valid bits shall be set to one, and the information field set to the total number of blocks, filemarks, or setmarks not spaced over (the requested number of blocks, filemarks, or setmarks minus the actual number of blocks, filemarks, or setmarks spaced over). A successfully completed SPACE command shall not set EOM to one at beginning-of-partition.

When spacing over sequential filemarks (or setmarks), 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 (or setmarks) being logically positioned after the N^{th} filemark (or setmark).
b) A zero value shall cause no change in the logical position.
c) A negative value -*N* (2's complement notation) shall cause reverse movement to the first occurrence of *N* or more consecutive filemarks (or setmarks) being logically positioned on the beginning-of-partition side of the N^{th} filemark (or setmark).

If a setmark is encountered while spacing to sequential filemarks and the RSmk bit is set to one in the device configuration page (see 5.3.3.2), CHECK CONDITION status shall be returned, the filemark bit shall be set to one, and the valid bit shall be set to zero. The sense key shall be set to NO SENSE and the additional sense code shall be set to SETMARK DETECTED. The device server shall not return CHECK CONDITION status when a setmark is encountered if the RSmk bit is set to zero or if this option is not supported.

If end-of-partition is encountered while spacing to sequential filemarks or setmarks, CHECK CONDITION status shall be returned, the sense key shall be set to MEDIUM ERROR, the EOM bit shall be set to one, and the valid bit shall be set to zero.

If end-of-data is encountered while spacing to sequential filemarks or setmarks, CHECK CONDITION status shall be returned, the sense key shall be set to BLANK CHECK, and the valid bit shall be set to zero. Additionally, the 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 logically recorded information.

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

5.2.13 VERIFY command

The VERIFY command (see table 21) requests that the device server verify one or more block(s) beginning with the next block on the logical unit.

Table 21 - VERIFY 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							

A reservation conflict shall occur when a VERIFY command is received from an initiator other than the one holding a logical unit reservation.

An immediate (Immed) bit of zero indicates that the command shall not return status until the verify operation has completed. An Immed bit of one indicates that status shall be returned as soon as the command descriptor block has been validated (but after all data has been transferred from the Data-Out buffer to the device server, if the BytCmp bit is one).

NOTE 18 In order to ensure that no errors are lost, the application client should set the Immed bit to zero on the last VERIFY command when issuing a series of VERIFY commands.

A byte compare (BytCmp) bit of zero indicates that 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 indicates that 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 command. 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 and the sense key shall be set to ILLEGAL REQUEST.

The verification length specifies the amount of data to verify, in blocks or bytes, as indicated by the fixed bit. Refer to the READ command (see 5.2.5) for a description of the fixed bit and any associated error conditions. If the BytCmp bit is one and the verification length 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 command shall terminate when the verification length has been satisfied, when an incorrect length block is encountered, when a filemark is encountered, when a setmark is encountered (if the RSmk bit is one in the device configuration page, see 5.3.3.2), when end-of-data is encountered, when the end-of-partition is encountered, when early-warning is encountered (if the REW bit is one in the device configuration page, see 5.3.3.2), or when an unrecoverable read error is encountered. The status and sense data for each of these conditions are handled in the same manner as in the READ command (see 5.2.5). Upon successful completion of a VERIFY command, the logical position shall be after the last block verified.

If the data does not compare (BytCmp bit of one), the command shall terminate with CHECK CONDITION status, the valid bit shall be set to one, and the sense key shall be set to MISCOMPARE. If the fixed bit is one, the information field shall be set to the requested verification length minus the actual number of 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 command with an Immed bit of one. Upon termination, the medium shall be positioned after the block containing the miscompare (end-of-partition side).

5.2.14 WRITE command

The WRITE command (see table 22) requests that the device server write the data that is transferred from the application client to the current position on the logical unit.

Table 22 - WRITE 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							

A reservation conflict shall occur when a WRITE command is received from an initiator other than the one holding a logical unit reservation.

The fixed bit specifies whether fixed-length or variable-length blocks are to be transferred. See the READ BLOCK LIMITS command (see 5.2.6) for additional information about fixed and variable block mode.

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 parameter block descriptor (see SPC). If the fixed bit is zero, a single block is transferred with the transfer length specifying the block length in bytes.

If the transfer length is zero, no data shall be transferred and the current position on the logical unit shall not be changed. This condition shall not be considered an error.

A WRITE command may be buffered or unbuffered, as indicated by the buffered mode field of the mode parameter header (see 5.3.3). For unbuffered operation (buffered mode 0h), the device server shall not return GOOD status until all data block(s) are successfully written to the medium. For buffered operation (buffered mode 1h or 2h), the device server may return GOOD status as soon as all data block(s) are successfully transferred to the logical unit's buffer.

NOTE 19 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 ensure that all buffered data, filemarks, and setmarks are written to the medium.

If the logical unit encounters early-warning during a WRITE 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 page (see 5.3.3.2). The command shall terminate with CHECK CONDITION status and the EOM and valid bits shall be set to one in the sense data. 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 any data that is to be written cannot be transferred to the medium when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW.

The information field shall be defined as follows:
a)If unbuffered mode is reported in the mode parameter header and the fixed bit is set to one, the information field shall be set to the requested transfer length minus the actual number of blocks written.
b)If unbuffered mode is reported and the fixed bit is set to zero, the information field shall be set to the requested transfer length.
c)If buffered mode is reported in the mode parameter header and the fixed bit is set to one, the information field shall be set to the total number of blocks, filemarks, and setmarks not written (the

number of blocks not transferred from the application client plus the number of blocks, filemarks, and setmarks remaining in the logical unit's buffer). Note that the value in the information field may exceed the transfer length.
d)If buffered mode is reported and the fixed bit is set to zero, the information field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of bytes not transferred from the application client plus the number of bytes, filemarks, and setmarks remaining in the logical unit's buffer).

NOTE 20 The value in the information field may exceed the transfer length.

NOTE 21 The logical unit should ensure that some additional data can be written to the medium (e.g. labels, filemarks, or setmarks) after the first early-warning indication has been returned to the application client.

If a WRITE command is received while the logical unit is positioned between early-warning and end-of-partition, the device server shall return CHECK CONDITION status after attempting to perform the command. The EOM and valid bits shall be set to one in the sense data. If all data that is to be written is successfully transferred to the medium, the information field shall be set to zero. If any data that is to be written is not transferred to the medium prior to encountering end-of-partition, the sense key shall be set to VOLUME OVERFLOW and the information field shall be defined as follows:
a) If the fixed bit is one, the information field shall be set to the requested transfer length minus the actual number of blocks written to the medium.
b) If the fixed bit is zero, the information field shall be set to the requested transfer length.

NOTE 22 In some systems it is important to recognize an error if end-of-partition is encountered during execution of a WRITE command, without regard for whether all data that is to be written is successfully transferred to the medium. By its definition, 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, as was done in some SCSI-1 implementations, may cause confusion as to whether there was really defective medium encountered during execution of the last write command.

If a WRITE command is terminated early, an incomplete logical block (a block not completely transferred to the device server from the Data-Out buffer) will be discarded by the logical unit. A subsequent WRITE command at the current logical position shall result in the loss of guaranteed access to the incomplete block. However, the logical unit does not guarantee that the incomplete block is totally unreadable until new data written at the current logical position is actually written to the media.

NOTE 23 Repositioning of the media may be required to remove a logical block from the media if the block is partially written before completely transmitted to the device server. While device-specific, a period of time may exist in which the partial block is not overwritten by the logical unit. Writing to the physical end of partition may also result in a partially written block. These blocks may result in a partial block transmission to the application client before the logical unit determines that the block is incomplete (bad).

5.2.15 WRITE FILEMARKS command

The WRITE FILEMARKS command (see table 23) requests that the device server write the specified number of filemarks or setmarks to the current position on the logical unit.

Table 23 - WRITE FILEMARKS command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (10h)							
1	Reserved						WSmk	Immed
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a WRITE FILEMARKS command is received from an initiator other than the one holding a logical unit reservation.

If the Write Setmark (WSmk) bit is one, the transfer length specifies the number of setmarks to be written. If the WSmk bit is zero, the transfer length specifies the number of filemarks to be written.

An immediate (Immed) bit of one indicates that 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 buffered mode is reported in the mode parameter header (see 5.3.3).

An Immed bit of zero indicates that the device server shall not return status until the write operation has completed. Any buffered data, filemarks, and setmarks shall be written to the medium prior to completing the command.

NOTE 24 Upon completion of any buffered write operation, the application client can issue a WRITE FILEMARKS command with the Immed bit set to zero and the transfer length field set to zero to ensure that all buffered data, filemarks, and setmarks are successfully written to the medium.

If the logical unit encounters early-warning during a WRITE FILEMARKS command, an attempt to finish writing any buffered data, filemarks, or setmarks may be made, as determined by the current settings of the REW and SEW bits in the device configuration page (see 5.3.3.2). The command shall terminate with CHECK CONDITION status and the EOM and valid bits shall be set to one in the sense data. If all buffered data, filemarks, and setmarks are successfully transferred to the medium, the sense key shall be set to NO SENSE or RECOVERED ERROR, as appropriate. If any buffered data, filemarks, or setmarks to be written are not transferred to the medium when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW.

The information field shall be defined as follows:

a) If unbuffered mode is reported in the mode parameter header, the information field shall be set to the requested transfer length minus the actual number of filemarks or setmarks written.
b) If buffered mode is reported in the mode parameter header and the buffered data was written in variable block mode (see 5.2.14), the information field shall be set to the total number of bytes, filemarks, and setmarks not written (the number of filemarks or setmarks not transferred from the application client plus the number of bytes, filemarks and setmarks remaining in the logical unit's buffer). Note that it is possible for the value in the information field to exceed the transfer length.
c) If buffered mode is reported and the buffered data was written in fixed block mode (see 5.2.14), the information field shall be set to the total number of blocks, filemarks, and setmarks not written (the number filemarks or setmarks not transferred from the application client plus the number of blocks, filemarks, and setmarks remaining in the logical unit's buffer). Note that it is possible for the value in the information field to exceed the transfer length.

NOTE 25 The logical unit should ensure that some additional data can be written to the medium (e.g. labels, filemarks, or setmarks) after the first early-warning indication has been returned to the application client.

If a WRITE FILEMARKS command is received while the logical unit is positioned between early-warning and end-of-partition, the device server shall return CHECK CONDITION status after attempting to perform the command. The EOM and valid bits shall be set to one in the sense data. If all filemarks or setmarks to be written are successfully transferred to the medium, the information field shall be set to zero. If any filemarks or setmarks to be written are not transferred to the medium prior to encountering end-of-partition, the sense key shall be set to VOLUME OVERFLOW and the information field shall be set to the requested transfer length minus the actual number of filemarks or setmarks written to the medium.

5.3 Parameters for sequential-access devices

5.3.1 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 24.

Table 24 - Diagnostic page codes

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

5.3.2 Log parameters

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 25.

Table 25 - Log page codes

Page Code	Description	Subclause
01h	Buffer over-run/under-run page	SPC
02h	Error counter page (write)	SPC
03h	Error counter page (read)	SPC
04h	Error counter page (read reverse)	SPC
05h	Error counter page (verify)	SPC
07h	Last *n* error events page	SPC
06h	Non-media error page	SPC
0Ch	Sequential-access device page	5.3.2.1
00h	Supported log pages	SPC
08h - 0Bh	Reserved	
0Dh - 2Fh	Reserved	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

5.3.2.1 Sequential-access device page

The sequential-access device page (page code 0Ch) 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 26 The data in parameters 0 and 1 are intended to provide an indication of the compression ratio for the written data. Parameters 2 and 2 are intended to provide an indication of the compression ratio for read data.

Support of this page is optional. Support of the individual parameters on this page are optional.

Table 26 defines the parameter codes for the sequential-access device page.

Table 26 - Parameter codes for sequential-access device 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 Data-Out buffers during READ command operations.
0004h - 00FFh	Reserved
0100h	Cleaning required
0101h - 7FFFh	Reserved
8000h - FFFFh	Vendor-specific parameters

A non-zero value of the cleaning required parameter indicates that a condition requiring cleaning has been detected and a subsequent cleaning cycle has not been completed. The cleaning required parameter shall be persistent across hard resets and power cycles.

5.3.3 Mode parameters

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.

The medium-type code field in the mode parameter header is vendor-unique for sequential-access devices.

The device-specific parameter byte of the mode parameter header (see SPC) is defined in table 27 for sequential-access devices.

Table 27 - Device-specific parameter

Bit	7	6	5	4	3	2	1	0
	WP	Buffered mode			Speed			

When used with the MODE SENSE command, a write protect (WP) bit of zero indicates that medium is write enabled. A WP bit of one indicates that the medium is currently write protected. When used with the MODE SELECT command, this field is not defined.

NOTE 27 Write-protect indicates that the medium is currently write-protected. The write-protect may be due to logical unit internal restrictions, soft write-protect, or a physical write-protect.

Values for the buffered mode field are defined in table 28.

Table 28 - Buffered modes

Code	Description
0h	The device server shall not report GOOD status on write commands until the data 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 buffer. One or more blocks may be buffered prior to writing the 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 buffer, and b) All buffered data from different application clients has been successfully written to the medium.
3h - 7h	Reserved

Values for the speed field shall be assigned as defined in table 29.

Table 29 - Speed field definition

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

For the MODE SELECT command, the density code field of the sequential-access device block descriptor (see SPC) indicates 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 installed 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. The principal density code is the highest density code supported, the optimal density code supported, or the most common density code supported. 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 as described below:

a) Following a UNIT ATTENTION condition for a power on or hard reset condition, while not ready, the device server shall report the principal density.
b) Following a UNIT ATTENTION condition for a not-ready-to-ready transition, the device server shall:
   1) report the principal density if no attempt has been made by the logical unit to determine the density;
   2) report the principal density if the logical unit cannot automatically determine the density from the medium;
   3) report the current medium density if the logical unit has determined the density from the medium.

c) Following a successful read operation at or after beginning-of-medium, 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 an unsuccessful read operation or a successful write operation, while at beginning-of-partition, the device server shall:

1) report a density code value as described for item b) if a previous MODE SELECT command has not established a density code for the currently mounted volume;
2) report a density code value as provided by the last successful MODE SELECT command for the currently mounted volume.

e) Following a successful unload operation the device server shall report the most recent density code value as determined by items b) through d) above.

For a MODE SELECT command, a density code of 7Fh shall indicate that the application client is not selecting a density. The value 7Fh shall not be returned by a MODE SENSE command. Table 30 lists the sequential-access device density codes.

Table 30 - Sequential-access density codes

Code Value	Description	Note
00h	Default density	1
01h-7Eh	Density code from REPORT DENSITY SUPPORT COMMAND	2
7Fh	No change from previous density (NO-OP)	3
80h-FFh	Density code from REPORT DENSITY SUPPORT COMMAND	

NOTES

1 Only reported by MODE SENSE if primary density code for the density.
2 See informative annex 8.1 for historical density codes.
3 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 for sequential-access devices are defined in table 31.

Table 31 - Mode page codes

Page Code	Description	Subclause
0Ah	Control mode page	SPC
0Fh	Data compression page	5.3.3.1
10h	Device configuration page	5.3.3.2
02h	Disconnect-reconnect page	SPC
1Ch	Log Exception Page	SPC
11h	Medium partition page(1)	5.3.3.3
12h	Medium partition page(2)	5.3.3.4
13h	Medium partition page(3)	5.3.3.4
14h	Medium partition page(4)	5.3.3.4
09h	Peripheral device page	SPC
01h	Read-write error recovery page	5.3.3.5
03h - 08h	Reserved	
0Bh - 0Fh	Reserved	
15h - 1Bh	Reserved	
1Dh - 1Fh	Reserved	
00h	Vendor-specific (does not require page format)	
20h - 3Eh	Vendor-specific (page format required)	
3Fh	Return all pages (valid only for the MODE SENSE command)	

5.3.3.1 Data Compression Page

The data compression page (see table 32) is used to specify the parameters for the control of data compression in a sequential-access device.

Table 32 - Data compression page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (0Fh)					
1	Page length (0Eh)							
2	DCE	DCC	Reserved					
3	DDE	RED		Reserved				
4 : 7	(MSB) Compression algorithm (LSB)							
8 : 11	(MSB) Decompression algorithm (LSB)							
12	Reserved							
13	Reserved							
14	Reserved							
15	Reserved							

A data compression enable (DCE) bit of one indicates that data compression is to be enabled. When this bit is set, 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 indicates that data compression is to be disabled.

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

A data decompression enable (DDE) bit of one indicates that data decompression is to be enabled. A DDE bit of zero indicates that data decompression is to be disabled.

The report exception on decompression (RED) field indicates the device's response to certain boundaries it detects in the data on the medium. There are a number of boundaries which may occur on the medium between compressed and uncompressed data. These boundaries are shown in table 33. Only boundaries shown in table may generate a CHECK CONDITION status.

Table 33 - Possible boundaries and resulting sense keys due to data compression

Prior Data	Current Data	Sense Key (see Note 1)		
		RED = 0	**RED = 1**	**RED = 2**
uncompressed	compressed (unsupported alg.)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
uncompressed	compressed (supported alg.)	[none]	[none]	RECOVERED ERROR
compressed (supported algorithm)	uncompressed	[none]	[none]	NO SENSE
compressed (supported algorithm)	compressed (unsupported alg.)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
compressed (supported algorithm A)	compressed (supported alg. B)	[none]	[none]	RECOVERED ERROR
compressed (unsupported algorithm)	uncompressed	[none]	NO SENSE	NO SENSE
compressed (unsupported algorithm)	compressed (supported alg.)	[none]	RECOVERED ERROR	RECOVERED ERROR
compressed (unsupported alg. A)	compressed (unsupported alg. B)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
Note 1:[none] indicates that no CHECK CONDITION status is returned given the data boundary condition and the current value of RED.				

A RED field of zero indicates that the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation which the device cannot decompress. Data boundaries in table marked other than [none] in the **RED = 0** shall generate CHECK CONDITION status with the specified sense key when the RED field is zero.

A RED field of one indicates that the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation which requires different handling by the application client than that data most recently encountered during a prior read operation. At each of these boundaries, the data which is sent to the application client is of a fundamentally different nature from that which was previously sent. Data boundaries in table marked other than [none] in the **RED = 1** shall generate CHECK CONDITION status with the specified sense key when the RED field is one.

A RED field of two indicates that the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation which has been processed using a different algorithm from that data most recently encountered during a prior read operation. Data boundaries in table marked other than [none] in the **RED = 2** shall generate CHECK CONDITION status with the specified sense key when the RED field is two.

A RED field of three is undefined and shall result in a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST.

On any of the boundary conditions described in table which 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 device shall be positioned on the EOP side of the encountered data, and the command-specific information field in the sense data shall contain a count of the number of data blocks contained within the encountered data.

NOTE 28 When compressed data is encountered on the medium which the device cannot decompress, the device should treat the data as a single variable-length record. In the sense data, the valid bit, the ILI bit and the information field should be set accordingly.

The compression algorithm field indicates 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 which the device does not support, then the device shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST. A value of zero shall indicate that no compression algorithm is currently selected. Algorithm identifiers are shown in table .

For the MODE SELECT command, the decompression algorithm field indicates 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, detected by the device, which 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, which was used to process the data most recently encountered on the medium, during a read operation. A value of zero shall indicate that the data encountered on the medium during the most recent read operation was uncompressed. Algorithm identifiers are shown in table .

Table 34 shows the compression algorithm identifiers registered under the International Register of Processing Algorithms [to be] established by ISO/IEC JTC1.

Table 34 - Compression algorithm identifiers

Algorithm Ident.	Description
00h	No algorithm selected (identifies uncompressed data).
01h	Unused
02h - 0Fh	Not assigned
10h	IBM IDRC data compaction algorithm
11h - 1Fh	Not assigned
20h	DCLZ data compression algorithm
21h - FDh	Not assigned
FEh	Reserved
FFh	Unregistered algorithm
100h - FFFFFFFFh	Reserved

5.3.3.2 Device configuration page

The device configuration page (see table 35) is used to specify the appropriate sequential-access device configuration.

Table 35 - Device configuration page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (10h)					
1	Page length (0Eh)							
2	Rsvd	CAP	CAF	Active format				
3	Active partition							
4	Write buffer full ratio							
5	Read buffer empty ratio							
6	(MSB) Write							
7	delay time (LSB)							
8	DBR	BIS	RSmk	AVC	SOCF		RBO	REW
9	Gap size							
10	EOD defined			EEG	SEW	SWP	Reserved	
11 : 13	(MSB) Buffer size at early warning (LSB)							
14	Select data compression algorithm							
15	Reserved							

A change active partition (CAP) bit of one indicates that the logical partition is to be changed to the one specified by the active partition field. If successful, the logical unit shall position to block 0 within the specified active partition. A CAP bit of zero indicates no partition change is specified.

A change active format (CAF) bit of one indicates that the active format is to be changed to the value specified in the active format field. A CAF bit of zero indicates 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 indicates which recording format 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 indicates the current logical partition number in use on the medium.

The write buffer full ratio, on WRITE commands, indicates to the logical unit how full the buffer shall be before writing data to the medium. A value of zero indicates that the value is not specified.

The read buffer empty ratio, on READ commands, indicates to the logical unit how empty the buffer shall be before retrieving additional data from the medium. A value of zero indicates that the value is not specified.

The write delay time indicates the maximum time, in 100 ms increments, that the logical unit 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 buffer to exceed the buffer full ratio. A value of zero indicates that the logical unit shall never force buffered data to the medium under these conditions.

A data buffer recovery (DBR) bit of one indicates that the logical unit supports data buffer recovery using the RECOVER BUFFERED DATA command. A DBR bit of zero indicates that the logical unit does not support data buffer recovery. This bit is target-defined.

A block identifiers supported (BIS) bit of zero indicates that block IDs are not supported in the format written on the medium. A BIS bit of one indicates that the format on the medium has recorded information about the block IDs relative to a partition. This bit is target-defined.

A report setmarks (RSmk) bit of one indicates that the device server shall recognize and report setmarks during appropriate read or space operations. A RSmk bit of zero indicates that the device server shall not report setmarks.

The automatic velocity control (AVC) bit of one, indicates that 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 indicates the speed chosen should be the device's default speed.

A stop on consecutive filemarks (SOCF) field of 00b indicates that the logical unit shall pre-read data from the medium in buffered mode to the limits of the buffer capacity without regard for filemarks. This implies that the device can differentiate between data blocks and filemarks in the buffer. Values 01b, 10b, and 11b specify that the device shall terminate the pre-read operation if one, two, or three consecutive filemarks are detected, respectively. If the RSmk bit is one, the logical unit shall interpret this field as stop on consecutive setmarks.

A recover buffer order (RBO) bit of one indicates that data blocks shall be returned from the logical unit's buffer on a RECOVERED BUFFERED DATA command in LIFO order (last-in-first-out) from which they were written to the buffer. A RBO bit of zero indicates data blocks shall be returned in FIFO (first-in-first-out) order.

A report early-warning (REW) bit of zero indicates that 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 indicates that the device server shall return CHECK CONDITION status with the EOM bit set 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 29 A REW bit of one is intended for compatibility with those systems using old tape formats that require an early-warning indication during read operations. Other systems should set this bit to zero to avoid potential data loss when interchanging tapes between peripheral devices.

The GapSize field value determines the size of the inter-block gap when writing data. A value of 00h specifies the device's defined gap size. A value of 01h specifies a device defined gap size sufficiently long to support update-in-place. Values of 02h through 0Fh are multipliers on the device's defined gap size. Values 10h through 7Fh are reserved. Values 80h through FFh are vendor-specific.

The end-of-data (EOD) defined field indicates which format type the logical unit shall use to detect and generate the EOD area. The types of EOD formats are specified in table 36.

Table 36 - EOD formats

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 indicates that the logical unit shall generate the appropriate EOD area, as determined by the EOD defined field. A value of zero indicates that EOD generation is disabled.

NOTE 30 Some logical units may not generate EOD at the completion of any write-type operation.

The synchronize at early-warning (SEW) bit set to one indicates that the logical unit shall cause any buffered write data, filemarks, or setmarks to be transferred to the medium when early-warning is encountered. A value of zero indicates that the logical unit shall retain any unwritten buffered data, filemarks, or setmarks in the buffer when early-warning is encountered (see the WRITE command, 5.2.13, and the WRITE FILEMARKS command, 5.2.14).

A soft write protect (SWP) bit of one indicates that the logical unit shall inhibit all writing to the medium after writing all buffered data, if any. When SWP is one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status with the additional sense code set to WRITE PROTECTED. A SWP bit of zero indicates that the logical unit may inhibit writing to the medium, dependent on other write inhibits.

The buffer size at early-warning field indicates the value, in bytes, to which the logical unit shall reduce its logical buffer size when writing. The logical unit should reduce the buffer size only when the logical unit is positioned between its early-warning and end-of-partition. A value of zero indicates that the implementation of this function is device specific.

NOTE 31 The intent is to prevent the loss of data by limiting the size of the buffer when near the end-of-partition.

The select data compression algorithm field set to 00h indicates that 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 indicates that 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.

5.3.3.3 Medium partition page(1)

The medium partition page(1) (see table 37) is used to specify the first group of medium partitions. Additional groups are specified in medium partition pages(2-4).

Table 37 - Medium partition page(1)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (11h)					
1	Page length							
2	Maximum additional partitions							
3	Additional partitions defined							
4	FDP	SDP	IDP	PSUM		Rsvd	CLEAR	ADDP
5	Medium format recognition							
6	Reserved				Partition units			
7	Reserved							
	Partition size descriptor(s)							
+ 0	(MSB) Partition							
+ 1	size (LSB)							

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 MODE SENSE indicates that 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 MODE SENSE 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 indicates that the logical unit shall partition the medium based on its fixed definition of partitions. Setting this bit to one 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 MODE SELECT when the FDP bit is set to one. The logical unit may assign any number of partitions from 1 to (maximum additional partitions + 1). Support for FDP set to one is optional.

NOTE 32 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 indicates that 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. The logical unit shall partition the medium into n+1 partitions numbered 0 through n. Setting this bit to one 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 MODE SELECT when the SDP bit is set to one. Support for SDP set to one is optional.

NOTE 33 The physical placement and order of medium partitions are not specified by this standard.

An initiator-defined partitions (IDP) bit of one indicates that the logical unit shall partition the medium as defined by the additional partitions defined field and the partition size descriptors. Setting this bit to one 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 medium partition pages(1-4) shall be one more than the additional partitions defined. The size of partition 0 shall be non-zero. Support for IDP set to one is optional.

A logical unit is not required to retain the method used to partition the medium. The MODE SENSE data shall return one and only one of the IDP, FDP or SDP fields set to one. If partitioned by FDP or SDP, a logical unit may return IDP set to one in the MODE SENSE data.

NOTE 34 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 any of the fields FDP, SDP, or IDP set to one.

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 38.

Table 38 - PSUM

Code	Description	Support
00b	bytes (unit of one)	Optional
01b	kilobytes (10^3 bytes)	Optional
10b	megabytes (10^6 bytes)	Optional
11b	$10^{(\text{Partition units field})}$ bytes	Optional

NOTE 35 The logical unit is not required to maintain the PSUM used to format any given media.

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 10N 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. The support of the Partition units field is optional.

NOTE 36 Some values of the Partition units field may result in no legal non-zero partition size descriptors.

A CLEAR bit of zero and an ADDP bit of zero indicate SCSI-2 compatibility. The logical unit may logically erase any or all partitions when one of the IDP, FDP, or SDP fields is set to one.

A CLEAR bit of one and an ADDP bit of zero indicates that 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 indicates that the logical unit shall not logically erase any existing partitions, even if the size of the partition is changed. If the MODE SELECT 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 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 partition size is equivalent to the current partition size, no change in the partition size shall result. If the logical unit cannot perform the operation or if such an operation would cause loss of valid data in any partition which exists both before and after the MODE SELECT, 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 with the additional sense code 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 plus one. An ADDP field set to zero indicates that the logical unit may logically erase any or all partitions if any of the IDP, SDP, or FDP bits is set to one.

If both the ADDP and CLEAR fields are set to one, the logical unit shall logically erase all partitions which 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 with the additional sense code 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 with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Support of the CLEAR bit set to one is optional. Support of the ADDP bit set to one is optional.

A MODE SELECT partition size descriptor has the equivalent (same) size as the current partition size if

1) The MODE SELECT size, PSUM, and Partition units fields are exactly the same as those returned by MODE SENSE or
2) The MODE SELECT size 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 the Partition units field or
3) The MODE SELECT size is 0FFFFh and the current size would return 0FFFFh if expressed in the units of the MODE SELECT PSUM or the Partition units field.

The medium format recognition field is a logical unit-defined value indicating the logical unit's capability to automatically identify the medium format and partition information when reading an unknown volume. The medium format recognition field values are shown in table 39.

Table 39 - Medium format recognition

Code	Description
00h	Logical unit is incapable of format or partition recognition.
01h	Logical unit is capable of format recognition only.
02h	Logical unit is capable of partition recognition only.
03h	Logical unit is capable of format and partition recognition.
04h - 0FFh	Reserved

NOTE 37 If a logical unit indicates that it is not capable of medium format recognition, the application client should supply all necessary parameters for the device to identify the specific format. The value in this field may be different following a medium change.

Partition size descriptors 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 can have size descriptors in this page. Partition N, if present, shall be described by the partition size descriptor at 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 page. The partition size descriptors for partitions 64 and greater can be defined in medium partition pages(2-4) defined in 1.1.1.2. 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 partition size descriptor size of 0FFFFh sent by MODE SELECT requests that the logical unit allocate all remaining partition space to that partition. A MODE SENSE shall return a partition size descriptor of 0FFFFh if the partition size, in units of PSUM or Partition units, is greater than or equal to 0FFFFh. If insufficient space exists on the medium for the requested partition sizes or if multiple partition size descriptors are set to 0FFFFh, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST. A device server may round, as described under Mode Select in SPC, any partition size to the nearest valid partition size.

NOTE 38 It is recommended, but not required, that the number of partition size descriptors available through medium partition pages(1-4) equal at least the number of maximum addition partitions + 1. This provides a mechanism for the device server to disclose the current partition sizes.

5.3.3.4 Medium partition page(2-4)

The medium partition page(2-4) (see table 40) is used to specify additional groups of medium partitions. The first group is specified in the medium partition page(1) (see 5.3.3.3).

Table 40 - Medium partition page(2-4)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (12h, 13h, 14h)					
1	Page length							
	Partition size descriptor(s)							
+ 0	(MSB) Partition							
+ 1	size (LSB)							

The additional page codes defined for partition size definition are 12h, 13h, and 14h. Up to a maximum of 64 partitions can be defined in each of these pages. The partitions size descriptors are numbered from N*64+0 to N*64+63 where N is equal to the medium partition page(2-4) minus 11h. The partition size descriptor for partition number P is located at bytes (P-N*64)*2+2 and (P-N*64)*2+3. The partition size is defined by the value of the partition size field. The units of size used by the partition size field is specified in the PSUM field of the medium partition page(1) (see 5.3.3.3).

Medium partition page(2) (page 12h) defines partitions numbered from 64 to 127.

Medium partition page(3) (page 13h) defines partitions numbered from 128 to 191.

Medium partition page(4) (page 14h) defines partitions numbered from 192 to 255.

If any of the medium partition pages(2-4) (pages 12h, 13h, and 14h) are supported, then each lower-numbered medium partition page shall be supported with the maximum length. Support of pages 12h, 13h, and 14h is not required if either:

1) The medium partition page defines only partitions which are invalid for the logical unit or
2) The logical unit does not support IDP set to one as defined in the medium partition page(1) (see 5.3.3.3).

NOTE 39 For a logical unit with N additional partitions, N+1 partitions may exist. Therefore, up to 63 additional partitions are supported by page 11h, and up to 128 partitions by both pages 11h and 12h. A maximum of 256 partitions are supported by pages 11h through 14h.

NOTE 40 It is recommended, but not required, that sufficient medium partition pages be supported to included all possible partitions when IDP is not supported. Support of medium partition pages (2-4) provides a mechanism for the device server to disclose the partition sizes.

NOTE 41 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.

If a MODE SELECT contains at least one of the medium partition pages(2-4) but does not contain medium partition page(1), the logical unit shall do one of the following:

a) Use the current values of PSUM, ADDP, and CLEAR to determine the method of partitioning. IDP is assumed to be set to one. If a conflict exists between ADDP and the partition size descriptors, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.
b) or the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

NOTE 42 The ADDP and CLEAR fields do not reflect the state of the media. These fields depend on settings provided by the application client.

If a MODE SELECT contains the medium partition page(1) and one or more of the medium partition pages(2-4), the logical unit shall process the data as one request. The logical unit shall not partition the medium more than once for a single MODE SELECT. If the application client sends duplicate medium partition pages, the logical unit shall use the last one of each medium partition page and ignore the partition size descriptors of the duplicated pages. The device server shall validate all medium partition page fields and return CHECK CONDITION status if any fields are invalid.

If a MODE SELECT contains the medium partition page(1) with the IDP field set to one and any of the supported medium partition pages(2-4) are not present in the Mode data, the logical unit shall:

a) If ADDP is set to zero, the logical unit shall define the partitions as given in medium partition page(1) with partitions greater than 63 defined as non-existent (zero length). The Addition Partitions Defined field is checked for legality as if the logical unit supported a maximum of 63 Additional Partitions.
b) If ADDP is set to one, the logical unit shall define the partitions as given in medium partition page(1) and the current values for medium partition pages (2-4). The Additional Partitions Defined field is checked for legality based on the number of non-zero partition size descriptors in the new medium partition page (1) and the existing medium partition pages (2-4).

NOTE 43 It is strongly suggested that Mode Select either send no medium partition pages or send all supported medium partition pages.

5.3.3.5 Read-write error recovery page

The read-write error recovery page (see table 41) specifies the error recovery and reporting parameters that the logical unit shall use when transferring data between the application client and the medium. These parameters only apply to read-write errors and do not affect message system retries or positioning error recovery procedures.

Table 41 - Read-write error recovery page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	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 - The parameters in this page also apply to verify operations.								

A transfer block (TB) bit of one indicates that a data block that is not recoverable within the specified recovery limits shall be transferred to the application client before CHECK CONDITION status is returned. A TB bit of zero indicates that the unrecoverable data block shall not be transferred to the

application client. Data blocks that can be recovered within the recovery limits are always transferred, regardless of the value of the TB bit.

An enable early recovery (EER) bit of one indicates that the logical unit shall use the most expedient error recovery algorithm (e.g. attempt error correction prior to retries). An EER bit of zero indicates that 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 block error-free prior to using error correction).

A post error (PER) bit of one indicates that the device server shall return CHECK CONDITION status to report recovered errors. A PER bit of zero indicates that 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 must also be set to zero.

A disable transfer on error (DTE) bit of one indicates that the device server shall terminate the data transfer after a recovered read-write error occurs. All data from the recovered block shall be transferred prior to terminating the read-write operation. A DTE bit of zero indicates that 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 indicates that 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 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 indicates that the logical unit shall not use its recovery algorithm during read operations.

The write retry count 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 indicates that the logical unit shall not use its recovery algorithm during write operations.

5.4 Definitions specific to sequential access devices

5.4.1. **beginning-of-partition:** The position at the beginning of the permissible recording region of a partition. If only one partition is defined, this position is typically equivalent to the beginning-of-medium.

5.4.2. **beginning-of-medium:** The extreme position along the medium in the direction away from the supply reel which can be accessed by the device.

5.4.3. **buffered mode:** A mode of data transfer in write operations which facilitates tape streaming (5.1.5), as reported in the mode parameter header device-specific parameter (5.3.3).

5.4.4. **early-warning:** A physical mark or device computed position near but logically before the end-of-partition (independent of physical direction). See the REW bit in the device configuration page in 5.3.3.2.

5.4.5. **end-of-data:** End of data in a partition is denoted in format-specific manner. See the EOD defined field in the device configuration page in 5.3.3.2.

5.4.6. **end-of-medium:** The extreme position along the medium in the direction away from the take-up reel which can be accessed by the device. This position may be accessed by devices that support the LOAD UNLOAD command with the EOT bit set to one.

5.4.7. **end-of-partition:** The position at the end of the permissible recording region of a partition.

5.4.8. **filemark:** A special recorded element within a partition, containing no user data, which provides a segmentation scheme.

5.4.9. **gap**: A non-data element recorded on the medium. Gaps may be recorded between logical elements. The format and method of recording a gap may vary.

5.4.10. **logical block**: A logical element which is a unit of data supplied or requested by an initiator.

5.4.11. **logical block address**: The logical block address identifies the logical element in a partition by an unique 4 byte value.

5.4.12. **logical element**: A unit of data, either a block or a mark. Each logical element has an unique logical block address, if supported, within the partition.

5.4.13. **mark**: A logical element which does not contain any initiator defined data. A mark is either a setmark or filemark.

5.4.14. **overlength:** The incorrect length condition that exists after executing a read command when the length of the actual block read exceeds the requested transfer length in the command descriptor block.

5.4.15. **partition:** The entire usable region of recording and reading paths in a volume or in a portion of a volume, defined in a device-specific manner. If there is more than one partition, they shall be numbered starting with zero (i.e. beginning-of-partition-zero).

5.4.16. **setmark:** A special recorded element within a partition, containing no user data, which provides a segmentation scheme hierarchically superior to filemarks for use in addressing or fast positioning on high capacity storage devices.

5.4.17. **spacing:** The act of positioning the medium on a sequential access device.

5.4.18. **tape:** Tape is the medium on which data is recorded. The medium is normally a long thin medium which is spooled onto one or two hubs, possibly within a cassette or cartridge.

5.4.19. **track:** A contiguous line on the medium consisting of a pattern of recorded signals writen by one write component.

5.4.20. **track group**: A set of tracks which are recorded at the same time.

5.4.21. **underlength:** The incorrect length condition that exists after executing a read command when the requested transfer length in the command descriptor block exceeds the length of the actual block read.

5.4.22. **volume:** A recording medium together with its physical carrier.

6 Printer devices

6.1 Model for printer devices

This command set includes capability for the printer-controlling device, that is an SCSI logical unit to be functionally separate from the physical printer device (see figure 11) as well as integrated with it. The physical printer device is connected to the SCSI target via one of several common device-level interfaces. There may be more than one physical printer device attached to the printer controlling device. In such a case, each physical printer device is assigned a separate logical unit number, beginning with zero. The printer-controlling device, printer device-level interface, and the physical printer device are referred to collectively as the printer device.

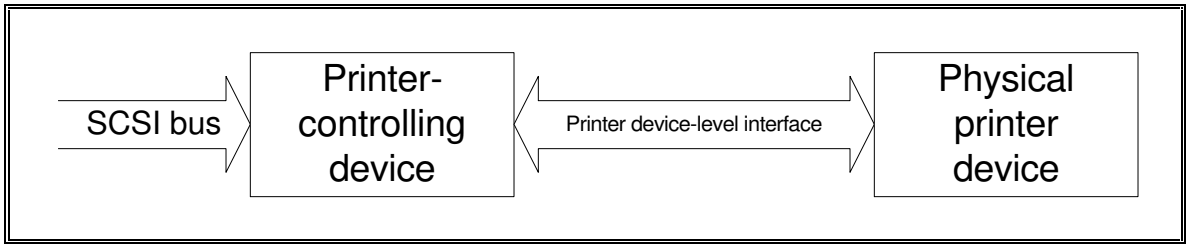

Figure 11 - SCSI printer model

Specific control mechanisms are defined in mode pages for two industry-standard interfaces known as the line printer interface (e.g. the Data Products interface or equivalent) and the EIA RS-232C interface. These mode pages are used to control optional features of these interfaces. No mode page was defined for the popular industry-standard parallel interface because the options requiring controls are embedded in the data.

The printer-controlling device may be integrated within the printer device; it is not required to use one of the industry-standard interfaces referenced above.

The printer device commands are structured on the assumption that specific printer control codes may be embedded in the data transferred by the FORMAT, PRINT, and SLEW AND PRINT commands. The transparent control codes may take the form of escape code sequences. commands for the operation of the logical unit function and some printer controls, which are not convenient to handle in a transparent way, are specified in 6.2.

This International Standard does not specify which character set is used by the printer device; nor does it specify the meaning of the escape code sequences that may be used.

The Reserve and Release commands (see SPC) are mandatory for printer devices. Extents are not supported by this model.

A printer device is ready when print commands can be executed. A printer is not ready if print media is unavailable for use, either from a lack of media or consumables, or by operator intervention (such as the off-line control). Such a device, with media unavailable, normally returns CHECK CONDITION status and sets the sense key to NOT READY.

6.2 Commands for printer devices

The commands for printer devices shall be as shown in table 42.

Table 42 - Commands for printer devices

Command Name	Operation Code	Type	Subclause
CHANGE DEFINITION	40h	O	SPC
COMPARE	39h	O	SPC
COPY	18h	O	SPC
COPY AND VERIFY	3Ah	O	SPC
FORMAT	04h	O	6.1.1
INQUIRY	12h	M	SPC
LOG SELECT	4Ch	O	SPC
LOG SENSE	4Dh	O	SPC
MODE SELECT (6)	15h	M	SPC
MODE SELECT (10)	55h	O	SPC
MODE SENSE (6)	1Ah	M	SPC
MODE SENSE (10)	5Ah	O	SPC
PRINT	0Ah	M	6.1.2
READ BUFFER	3Ch	O	SPC
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	SPC
RECOVER BUFFERED DATA	14h	O	6.1.3
RELEASE UNIT	17h	M	SPC
REQUEST SENSE	03h	M	SPC
RESERVE UNIT	16h	M	SPC
SEND DIAGNOSTIC	1Dh	M	SPC
SLEW AND PRINT	0Bh	O	6.1.4
STOP PRINT	1Bh	O	6.1.5
SYNCHRONIZE BUFFER	10h	O	6.1.6
TEST UNIT READY	00h	M	SPC
WRITE BUFFER	3Bh	O	SPC

Key: M = command implementation is mandatory.
O = command implementation is optional.
SPC = SCSI-3 Primary Command Set Standard

The following operation codes are vendor-specific: 01h, 02h, 05h, 06h, 07h, 08h, 09h, 0Ch, 0Dh, 0Eh, 0Fh, 11h, 13h, 19h, and C0h through FFh. All remaining operation codes for printer devices are reserved for future standardization.

6.1.1 FORMAT command

The FORMAT command (see table 43) provides a means for the application client to specify forms or fonts to printers that support programmable forms or fonts. The format information is peripheral-device specific.

Table 43 - FORMAT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (04h)							
1	Reserved						Format Type	
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a FORMAT command is received from an initiator other than the one holding a logical unit reservation.

The format type field specifies the type of format information to be transferred from the application client to the logical unit. This field is defined in table 44.

Table 44 - Format type

Code	Format type	Support
00b	Set form	Optional
01b	Set font	Optional
10b	Vendor-specific	
11b	Reserved	

The transfer length specifies the length in bytes of format information that shall be located in the Data-Out buffer. A transfer length of zero indicates that no format information shall be transferred. This condition shall not be considered an error.

6.1.2 PRINT

The PRINT command (see table 45) transfers the specified number of bytes from the application client to the device server to be printed.

Table 45 - PRINT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (0Ah)							
1	Reserved							
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a PRINT command is received from an initiator other than the one holding a logical unit reservation.

The transfer length specifies the length in bytes of data that shall be located in the Data-Out buffer. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as error.

6.1.3 RECOVER BUFFERED DATA command

The RECOVER BUFFERED DATA command (see table 46) returns to the application client data that was transferred to the device server, but not yet printed.

Table 46 - RECOVER BUFFERED DATA command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (14h)							
1	Reserved							
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a RECOVER BUFFERED DATA command is received from an initiator other than the one holding a logical unit reservation.

This command is normally used only to recover from error or exception conditions that make it impossible to print the buffered data. The order in which data are transferred from the device server to the application client is the same as when the data were previously transferred using the PRINT command or SLEW AND PRINT command. Data transferred by this command are deleted from the logical unit data buffer. One or more RECOVER BUFFERED DATA commands may be used to return the buffered data that is not printed.

If an attempt is made to recover more data than is contained in the buffer, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to NO SENSE. In addition, the EOM, valid, and ILI bits shall be set to one. The information bytes shall be set to the difference (residue) between the transfer length and the actual number of bytes returned.

The transfer length specifies the maximum length in bytes of data that shall be transferred to the Data-In buffer. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as error.

6.1.4 SLEW AND PRINT command

The SLEW AND PRINT command (see table 47) transfers the specified number of bytes from the application client to the device server to be printed. This command is provided for printer devices that do not support forms control information embedded within the print data.

Table 47 - SLEW AND PRINT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (0Bh)							
1	Reserved							Chanl
2	Slew value							
3	(MSB)							
4	Transfer Length							(LSB)
5	Control							

A reservation conflict shall occur when a SLEW AND PRINT command is received from an initiator other than the one holding a logical unit reservation.

The transfer length specifies the length in bytes of data that shall be located in the Data-Out buffer. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered as error.

If the channel (Chanl) bit is zero, the slew value specifies the number of lines the form shall be advanced before printing. A value of 255 indicates that the form shall be advanced to the first line of the next form before printing. If the channel bit is one, the slew value specifies the forms control channel number to which the form shall be advanced prior to printing the data. If implemented, the printer options page in the MODE SELECT command provides additional control over the usage of the slew value field.

If the channel bit is one, and the channel option is not implemented, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

6.1.5 STOP PRINT command

The STOP PRINT command (see table 48) requests that the logical unit halt printing on buffered devices in an orderly fashion.

Table 48 - STOP PRINT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (1Bh)							
1	Reserved							Retain
2	Vendor-specific							
3	Reserved							
4	Reserved							
5	Control							

A reservation conflict shall occur when a STOP PRINT command is received from an initiator other than the one holding a logical unit reservation.

A retain bit of zero requests that the logical unit data buffer be discarded; otherwise, the data not printed shall be retained. The data not printed may be recovered using the RECOVER BUFFERED DATA command, if implemented by the logical unit. A subsequent SYNCHRONIZE BUFFER command, PRINT command, or SLEW AND PRINT command shall cause the remaining data not already printed to be printed, followed by the data transferred by the subsequent command, if any. The point at which printing is suspended by this command is vendor-specific.

6.1.6 SYNCHRONIZE BUFFER command

The SYNCHRONIZE BUFFER command (see table 49) provides a means for an application client to ensure that the data have been printed successfully prior to releasing the peripheral device. This is useful for applications that handle any error or exception conditions (e.g. end-of-medium) prior to termination of the application.

Table 49 - SYNCHRONIZE BUFFER command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (10h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Control							

A reservation conflict shall occur when a SYNCHRONIZE BUFFER command is received from an initiator other than the one holding a logical unit reservation.

When all buffered data are actually printed, the command shall be terminated with a GOOD status. If it is not possible to finish printing all of the buffered data (due to an error or exception condition on the peripheral device), then this command shall be terminated with a CHECK CONDITION status and the appropriate sense key.

The printer options page in the MODE SELECT command, if implemented, provides additional control over termination sequences when using this command.

6.2 Parameters for printer devices

6.2.1 Diagnostic parameters

This subclause defines the descriptors and pages for diagnostic parameters used with printer devices.

The diagnostic page codes for printer devices are defined in table 50.

Table 50 - Diagnostic page codes

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

6.2.2 Log parameters

This subclause defines the descriptors and pages for log parameters used with printer devices.

The log page codes for printer devices are defined in table 51.

Table 51 - Log page codes

Page Code	Description	Subclause
01h	Buffer over-run/under-run page	SPC
07h	Last *n* error events page	SPC
06h	Non-media error page	SPC
00h	Supported log pages	SPC
02h - 05h	Reserved	
08h - 2Fh	Reserved	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

6.2.3 Mode parameters

This subclause defines the descriptors and pages for mode parameters used with printer devices.

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

The medium-type code field is contained in the mode parameter header (see SPC). This field is reserved for printer devices.

The device-specific parameter field is contained in the mode parameter header (see SPC). Table 52 defines the device-specific parameter used for printer devices.

Table 52 - Printer device-specific parameter

Bit	7	6	5	4	3	2	1	0
	WP	Buffered mode			Reserved			

When using the MODE SELECT command, a buffered mode of zero indicates that the device server shall not report a GOOD status on PRINT commands or SLEW AND PRINT commands until the data are actually printed. A buffered mode of one indicates that the device server may report a GOOD status on PRINT commands or SLEW AND PRINT commands as soon as the data have been transferred to the SCSI device buffer. The data from one or more commands may be buffered prior to printing. Buffered modes of 2h through 7h are reserved.

When using the MODE SENSE command, the buffered mode field returns the current value of this parameter.

The density code field is contained in the mode parameter block descriptor (see SPC). This field is reserved for printer devices.

The mode page codes for printer devices are shown in table 53.

Table 53 - Mode page codes

Page Code	Description	Subclause
0Ah	Control mode page	SPC
02h	Disconnect-reconnect page	SPC
03h	Parallel printer interface page	6.2.3.1
09h	Peripheral device page	SPC
05h	Printer options page	6.2.3.2
04h	Serial printer interface page	6.2.3.3
01h	Reserved	
06h - 08h	Reserved	
0Bh - 1Fh	Reserved	
00h	Vendor-specific (does not require page format)	
20h - 3Eh	Vendor-specific (page format required)	
3Fh	Return all pages (valid only for the MODE SENSE command)	

6.2.3.1 Parallel printer interface page

The parallel printer interface page (see table 54) is intended to support printer devices that use the industry-standard line printer interface.

Table 54 - Parallel printer interface

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (03h)					
1	Page length (03h)							
2	Parity select		PIPC	Rsvd	VCBP	VCBS	VEC	Autofd
3	Reserved							

The parameters savable (PS) bit is only used with the MODE SENSE command. This bit is reserved with the MODE SELECT command. A PS bit of one indicates that the logical unit is capable of saving the page in a non-volatile, vendor-specific location.

The parity select field specifies parity generation on the printer interface is defined in table 55.

Table 55 - Parity select

Code	Description
00b	No parity generation
01b	Even parity
10b	Odd parity
11b	Reserved

A paper instruction parity check (PIPC) bit of one indicates that the printer interface Paper Instruction signal is included in parity generation on the printer interface by the logical unit. A PIPC bit of zero indicates that the Paper Instruction signal is not included in parity generation on the printer interface by the logical unit.

NOTE 44 The format of the vertical forms unit (VFU) control byte is specified by the VCBP, VCBS, and VES bits. The VFU control byte is not part of this standard; however, a discussion of it is included here for a better understanding. This VFU control byte (see table, below) includes a control bit (C) to select whether to slew over a number of lines or to skip to a vertical forms unit (VFU) channel number. This bit may be located in two different bit positions and may have either polarity.

NOTE 45 The number of lines to slew over may be coded as an unsigned four-bit number (NNNN) or an unsigned six-bit number (EENNNN). The upper two bits of the six-bit number (EE) may be positioned contiguously with the lower four bits (NNNN) or they may be separated from the lower four bits by the control bit (C). Furthermore, the upper two bits may or may not be supported.

NOTE 46 The VFU channel to select is specified by an unsigned four-bit number (NNNN) in the VFU control byte.

NOTE 47 Pictorially, the control byte is shown in table 56.

Table 56 - VFU control byte

VFU control byte								Description	
Bits 7	6	5	4	3	2	1	0		
0	E	E	C	N	N	N	N	Advance (EE)NNNN lines	(C = line skip)
0	0	0	C	N	N	N	N	Skip to VFU channel NNNN	(C = VFU channel)
0	C	E	E	N	N	N	N	Advance (EE)NNNN lines	(C = line skip)
0	C	0	0	N	N	N	N	Skip to VFU channel NNNN	(C = VFU channel)
Key: C = VFU control bit									

A VFU control bit polarity (VCBP) bit of one indicates that the VFU control bit (C) is true for a VFU channel command and false for a line skip command. A VCBP bit of zero indicates the opposite polarity.

A VFU control bit select (VCBS) bit of one indicates that the control bit (C) is in bit position 6 and the EE bits are in bit positions 5 and 4. A VCBS bit of zero indicates that the control bit (C) is in bit position 4 and the EE bits are in bit positions 6 and 5.

A VFU extended skip (VES) bit of one indicates that the EE bits are supported for line skip operations. A VES bit of zero indicates that the EE bits are not supported.

For MODE SELECT, an automatic line feed (Autofd) bit of one specifies that the logical unit assert the printer interface Auto Line Feed signal. An Autofd bit of zero specifies that the logical unit negate the printer interface Auto Line Feed signal. For MODE SENSE, the device server shall return the current value of this bit.

6.2.3.2 Printer options page

The printer options page (see table 57) supports control and reporting of various logical unit functions and features.

Table 57 - Printer options

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (05h)					
1	Page length (0Ah)							
2	EVFU	Font identification						
3	Reserved		Slew mode		Reserved		SCTE	AFC
4	(MSB) Maximum							
5	line length (LSB)							
6	EVFU format start character							
7	EVFU format stop character							
8	Line slew options				Form slew options			
9	Data termination options				Reserved			
10	Reserved							
11	Reserved							

The parameters savable (PS) bit is only used with the MODE SENSE command. This bit is reserved with the MODE SELECT command. A PS bit of one indicates that the logical unit is capable of saving the page in a non-volatile vendor-specific location.

An electronic vertical forms unit (EVFU) bit of one indicates that a tape or electronic vertical forms unit is present in the printer device. An EVFU bit of zero indicates that the electronic vertical forms unit should not be used or that the unit is not present in the printer device.

For the MODE SELECT command, the font identification field specifies which font is to be used. For the MODE SENSE command, this field identifies the currently selected font. The font identification codes are defined table 58.

Table 58 - Font identification

Value	Description
00h	Default font
01h - 3Fh	Reserved
40h - 7Fh	Vendor-specific font

For the MODE SELECT command, the slew mode field controls the logical unit's behavior when a SLEW AND PRINT command is received with a channel bit of zero (see 6.1.4). For the MODE SENSE command, this field reports the current slew mode. This slew mode is defined in table 59.

Table 59 - Slew mode

Code	Description
00b	SLEW AND PRINT commands with a channel bit of zero are supported without any required setup.
01b	SLEW AND PRINT commands with a channel bit of zero are only accepted after the application client issues a FORMAT command with a format type of set form (to initialize the electronic vertical forms unit).
10b	SLEW AND PRINT commands with a channel bit of zero are always accepted. The logical unit shall use the electronic vertical forms unit if it has previously been initialized or it shall initialize the electronic vertical forms unit for line skipping by sending the EVFU start format character immediately followed by the EVFU stop format character (both are defined below) prior to executing the SLEW AND PRINT command.
11b	Reserved

For the MODE SELECT command, a step count truncate enable (SCTE) bit of one indicates that the logical unit shall assume the printer truncates slew values in SLEW AND PRINT commands, so that the data prints on the first line of the next form. A SCTE bit of zero indicates that the logical unit shall assume the printer continues to slew over forms boundaries in SLEW AND PRINT commands if the slew value exceeds the number of remaining lines on the current form. The device server shall return the current value of this field in the MODE SENSE data.

An ASCII forms control (AFC) bit of one indicates that the printer supports ASCII forms control characters. An AFC bit of zero indicates that the printer does not support ASCII forms control characters.

The maximum line length field specifies the maximum transfer length (maximum number of bytes per line) to be accepted in the SLEW AND PRINT command. A value of 0000h in MODE SELECT specifies that the logical unit shall use its default value. In any case, the device server shall report its actual value in the MODE SENSE data (not 0000h).

The EVFU format start character field specifies the character code to be used by the logical unit to start the initialization of the electronic vertical forms unit, if slew mode option 10b is selected.

The EVFU format stop character field specifies the character code to be used by the logical unit to stop the initialization of the electronic vertical forms unit, if slew mode option 10b is selected.

The line slew options field specifies the implementation of the line slew (using ASCII forms control characters) in the SLEW AND PRINT command. Code values in this field are defined in table 60.

Table 60 - Line slew

Code	Description
0h	Not implemented. (SLEW AND PRINT commands cause CHECK CONDITION status with ILLEGAL REQUEST sense key.)
1h	The logical unit shall insert an ASCII carriage return character (0Dh) for each line to slew over.
2h	The logical unit shall insert an ASCII line feed character (0Ah) for each line to slew over.
3h	The logical unit shall insert an ASCII carriage return character (0Dh) and line feed character (0Ah) for each line to slew over.
4h - 7h	Reserved
8h - Fh	Vendor-specific

The form slew options field specifies the implementation of form slewing in the SLEW AND PRINT command. Code values in this field are defined in table 61.

Table 61 - Form slew

Code	Description
0h	Not implemented. (SLEW AND PRINT commands cause CHECK CONDITION status with ILLEGAL REQUEST sense key.)
1h	The logical unit shall insert an ASCII form feed character (0Ch) to move to the beginning of the next form.
2h	The logical unit shall insert an ASCII carriage return character (0Dh) and form feed character (0Ch) to move to the beginning of the next form.
3h - 7h	Reserved
8h - Fh	Vendor-specific

The data termination options field specifies the termination sequence to be issued to the printer device when a SYNCHRONIZE BUFFER command is received. Code values for this field are defined in table 62.

Table 62 - Data termination option

Code	Description
0h	Selects the logical unit default implementation (MODE SELECT).
1h	No termination sequence. (The logical unit sends any remaining data in its buffer to the printer device with no termination sequence.
2h	The logical unit shall print any buffered data followed by an ASCII carriage return character (0Dh).
3h	The logical unit shall print any buffered data followed by an ASCII line feed character (0Ah).
4h	The logical unit shall print any buffered data followed by an ASCII carriage return, line feed character sequence (0Dh, 0Ah).
5h	The logical unit shall send any buffered data followed by an ASCII form feed character (0Ch).
6h	The logical unit shall print any buffered data followed by an ASCII carriage return, form feed character sequence (0Dh, 0Ch).
7h	The logical unit shall issue a zero line slew command to the printer device.
8h - Bh	Reserved
Ch - Fh	Vendor-specific

6.2.3.3 Serial printer interface page

The serial printer interface page (see table 63) is intended to support printer devices that use the industry-standard serial interface usually referred to as EIA RS-232C.

Table 63 - Serial printer interface

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Rsvd	Page Code (04h)					
1	Page length (06h)							
2	Reserved		Stop bit length					
3	Parity selection			Rsvd	Bits per character			
4	RTS	CTS	Reserved		Pacing protocol			
5	(MSB)							
6	Baud rate							
7	(LSB)							

The parameters savable (PS) bit is only used with the MODE SENSE command. This bit is reserved with the MODE SELECT command. A PS bit of one indicates that the logical unit is capable of saving the page in a non-volatile vendor-specific location.

The stop bit length field specifies the length of the stop bit(s) in units of 1/16 bit. The device server may round this field as described in SPC. A stop bit length of zero in the MODE SELECT command indicates that the logical unit shall use its default value.

The parity selection field specifies parity generation and checking as defined in table 64.

Table 64 - Parity selection

Code	Description
000b	None
001b	Mark
010b	Space
011b	Odd
100b	Even
101b-111b	Reserved

The bits per character field specifies the number of bits in each character. A value of zero in the MODE SELECT command indicates that the logical unit shall use its default character size.

For the MODE SELECT command, a request to send (RTS) bit of one specifies that the logical unit shall insure the Request to Send (RTS) signal in the serial interface follows the line state of the Data Terminal Ready (DTR) signal in the serial interface. A RTS bit of zero specifies that the logical unit shall set the Request to Send (RTS) signal to high whenever the logical unit power is on. For the MODE SENSE command, the device server shall report the current value of this field.

For the MODE SELECT command, a clear to send (CTS) bit of one specifies that the logical unit shall delay data transmission to the printer device until the printer device asserts the Clear to Send (CTS) signal in the serial interface. A CTS bit of zero indicates that the logical unit shall ignore the Clear to Send signal. For the MODE SENSE command, the device server shall report the current value of this field.

The code values for the pacing protocol field are defined in table 65.

Table 65 - Pacing protocol

Code	Description
0h	None
1h	XON/XOFF
2h	ETX/ACK
3h	DTR
4h - 7h	Reserved
8h - Fh	Vendor-specific

The baud rate field specifies the baud rate in bits per second. The device server may round this value as described in SPC. A baud rate of zero in the MODE SELECT command specifies that the logical unit shall use its default baud rate.

7 Communications devices

7.1 Communications device model

A communications device provides a facility to send and receive information over some medium (usually an electrical or fiber-optic cable) using a defined protocol to one or more other systems that support the same protocol. The media and protocols are often specified in national or international standards, although some are proprietary. Within this subclause, the protocol used on the external medium is called the external protocol.

The SCSI communications device model assumes that information to control the external protocol is embedded within the data transferred by the SEND and GET MESSAGE commands. The structure of the data transferred by these commands is not defined by this International Standard. Vendor-unique MODE pages may be defined to provide additional control over the external protocol.

The SCSI communications device is a logical unit with the characteristics of a network access device, typically attached to a local area network (LAN), public telecommunications network, private telecommunications network, or packet switching network, etc., with no theoretical limits. Such a communications device is often called a network node. The communications device transmits or receives packets of data as requested by the application client. The contents and meaning of the data packets is not defined by this standard. The bytes in the data packets may or may not contain addressing, path selection, or path control information identifying separate data streams. Additional information may be provided by the stream selection identifier found in the ten and twelve byte SEND MESSAGE and GET MESSAGE commands.

In the SCSI communications device, the logical unit accepts and provides the data packets transferred in accordance with commands received from the application client. There is an assumption that the initiator and the communications device know the rules by which information is exchanged between them, how the information is to be interpreted by the communications device, and when it is allowable to exchange the information. These rules are not specified by this International Standard.

The application client requests that the communications device accept a packet of data by transmitting a SEND MESSAGE command. The application client requests that the communications device return a packet of data by transmitting a GET MESSAGE command. The communications device also supports commands related to configuration of the network access device and network management of the particular environment. MODE SENSE and MODE SELECT are examples of these configuration commands.

If a communications device temporarily has no resource available to handle a data packet from the application client, or has no data packet to provide to the application client, or has no resources assigned to perform the operation, the device may then choose one of the following responses:

a) Terminate the command with CHECK CONDITION status. The sense key shall be set to NOT READY and the appropriate additional sense code for the condition.
b) Disconnect until the necessary resource or data packet becomes available, and then reconnect to the application client and resume the operation.
c) Terminate the command with BUSY status.

More than one logical unit can be implemented by a communications device. Logical units can serve as multiple paths to a single network access device, and/or each logical unit can serve as a discrete path to different resources within the device. A single logical unit may also serve as a path to multiple resources if the communications device can interpret information within the data packet and route the packet to the

appropriate resource. If the logical unit addressed by the application client does not have an available resource or data packet associated with it, the communications device may choose to treat the logical unit as an invalid logical unit (see SPC) or respond as described in the previous paragraph.

If the communications device determines that an error or unusual condition has occurred while performing an operation specified by the contents of a data packet, the information describing the condition is normally returned as a part of a data packet from another network device. If the communications device determines that an error or unusual condition has occurred while either executing the SCSI command from the application client, or during a network medium access transaction, the command is terminated with a CHECK CONDITION. The failures are identified through a REQUEST SENSE command.

The SCSI communications device is distinguished from an SCSI processor device by the fact that the primary destination of the data packets transferred to the communications device is not the target device itself, but another network node. A SCSI communications device passes the data on to an ultimate destination outside the target through a network. In contrast, the SCSI processor device is itself the primary destination of the data packets. Devices requiring protocols and command sets that are totally incompatible with the communications device protocols should be examined carefully to be sure that the incompatibilities are based on functional requirements.

The Reserve and Release commands (see SPC) are mandatory for communication devices. Extents are not supported by this model.

A communications device is ready when send commands can be executed. A communications device is not ready if the device cannot accept packets of data from the application client if successful transmission of the data is not expected (such as when the physical connection is broken) or insufficient resources are available to handle new requests. Such a device normally returns CHECK CONDITION status and sets the sense key to NOT READY.

7.1.1 Implementation examples

Several examples of communications device implementations are provided to clarify the range of utility of this device type.

7.1.1.1 Host-to-host communications

In this example of host to host communications, only the SEND MESSAGE command is used.

A communications device (comm A) is attached to a network (net A), and another communications device (comm B) is attached to a network (net B). Comm A takes the initiator role and selects a communications device (comm B), transmitting a packet to comm B using the SEND MESSAGE command. After transmitting the packet on the attached network medium (net B), the action required by the SEND MESSAGE command, comm B assumes the initiator mode and selects comm A as a communications device, and uses a SEND MESSAGE command to transmit a packet on the network (net A) attached to comm A.

NOTE 48 The SEND MESSAGE command is sufficient to perform complete transactions between communications devices if comm A and comm B act as initiators. This provides the function of a network bridge or gateway for high bandwidth intercommunication among nearby host processors.

7.1.1.2 Host-to-device communications

In this example of host to device communications, the SEND MESSAGE and GET MESSAGE commands are used.

A host system, host A, takes the initiator role and selects a communications device (comm A) attached to a network (net A), transmitting a packet on network net A to some other network node(s) using the SEND MESSAGE command to comm A. Host A then assumes that a result will be obtained by the other network node(s) consistent with rules understood by all involved network devices. Host A generates a GET MESSAGE command to obtain packets from other network nodes. If there are no packets that need handling pending at comm A, comm A may disconnect until a packet arrives. Comm A can then complete the transaction, reconnect to host A and the requested data packet can be returned to host A. Note that Host A need not support logical unit mode and Comm A need not support initiator mode to successfully complete an exchange between the two devices. The host system (host A) can be replaced by a communications device that is capable of acting as an initiator.

7.1.1.3 Multiple role communications

In this example of multiple role communications, the SEND MESSAGE and GET MESSAGE commands are used. The device acts as both a host and a communications device, depending on its needs and the requests made in the network.

A communications device (comm A) is attached to a network (net A), and another communications device (comm B) is attached to a network (net B). Comm A takes the initiator role and selects a communications device (comm B). Comm A transmits a packet for network net B using the SEND MESSAGE command to comm B. Comm A then requests data packets from net B by issuing a GET MESSAGE command to the comm B device. Following these actions, comm B assumes the initiator role and transmits a packet using the SEND MESSAGE command to comm A for transmission to network node(s) located on net A. Comm B then requests data packets from net A by issuing a GET MESSAGE command to comm A.

7.2 Command descriptions for communications devices

The commands for communications devices shall be as shown in table 66.

Table 66 - Commands for communications devices

Command Name	Operation Code	Type	Subclause
CHANGE DEFINITION	40h	O	SPC
GET MESSAGE(6)	08h	O	7.2.1
GET MESSAGE(10)	28h	O	7.2.2
GET MESSAGE(12)	A8h	O	7.2.3
INQUIRY	12h	M	SPC
LOG SELECT	4Ch	O	SPC
LOG SENSE	4Dh	O	SPC
MODE SELECT (6)	15h	M	SPC
MODE SELECT (10)	55h	O	SPC
MODE SENSE (6)	1Ah	M	SPC
MODE SENSE (10)	5Ah	O	SPC
READ BUFFER	3Ch	O	SPC
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	SPC
REQUEST SENSE	03H	M	SPC
RELEASE UNIT	17h	M	SPC
RESERVE UNIT	16h	M	SPC
SEND DIAGNOSTIC	1Dh	M	SPC
SEND MESSAGE(6)	0Ah	M	7.2.4
SEND MESSAGE(10)	2Ah	O	7.2.5
SEND MESSAGE(12)	AAh	O	7.2.6
TEST UNIT READY	00h	M	SPC
WRITE BUFFER	3Bh	O	SPC
Key: M = command implementation is mandatory. O = command implementation is optional. SPC = SCSI-3 Primary Command Set Standard			

All other operation codes for communications devices are reserved for future standardization.

7.2.1 GET MESSAGE(6) command

The GET MESSAGE(6) command (see table 67) transfers data from the device server to the application client.

Table 67 - GET MESSAGE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (08h)							
1	Reserved							
2	(MSB)							
3	Allocation Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a GET MESSAGE(6) command is received from an initiator other than the one holding a logical unit reservation.

7.2.2 GET MESSAGE(10) command

The GET MESSAGE(10) command (see table 68) transfers data from the device server to the application client.

Table 68 - GET MESSAGE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (28h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	(MSB)			Stream				
5				selection				(LSB)
6	Reserved							
7	(MSB)			Allocation				
8				length				(LSB)
9	Control							

A reservation conflict shall occur when a GET MESSAGE(10) command is received from an initiator other than the one holding a logical unit reservation.

The stream selection field specifies a further level of addressing for the data, so that it can be accessed by the logical unit from the appropriate data stream.

7.2.3 GET MESSAGE(12) command

The GET MESSAGE(12) command (see table 69) transfers data from the device server to the application client.

Table 69 - GET MESSAGE(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (A8h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	(MSB)			Stream				
5				selection				(LSB)
6	(MSB)							
7				Allocation				
8				length				
9								(LSB)
10	Reserved							
11	Control							

A reservation conflict shall occur when a GET MESSAGE(12) command is received from an initiator other than the one holding a logical unit reservation.

The stream selection field specifies a further level of addressing for the data, so that it can be accessed by the logical unit from the appropriate data stream.

7.2.4 SEND MESSAGE(6) command

The SEND MESSAGE(6) command (see table 70) transfers data from the application client to the device server.

Table 70 - SEND MESSAGE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (0Ah)							
1	Reserved							
2	(MSB)							
3	Transfer Length							
4								(LSB)
5	Control							

A reservation conflict shall occur when a SEND MESSAGE(6) command is received from an initiator other than the one holding a logical unit reservation.

The transfer length specifies the length in bytes of data that shall be located in the Data-Out buffer. A transfer length of zero indicates that no data shall be sent. This condition shall not be considered an error.

7.2.5 SEND MESSAGE(10) command

The SEND MESSAGE(10) command (see table 71) transfers data from the application client to the device server.

Table 71 - SEND MESSAGE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Ah)							
1	Reserved							
2	Reserved							
3	Reserved							
4	(MSB)			Stream				
5				selection				(LSB)
6	Reserved							
7	(MSB)			Transfer				
8				length				(LSB)
9	Control							

A reservation conflict shall occur when a SEND MESSAGE(10) command is received from an initiator other than the one holding a logical unit reservation.

The stream selection field specifies a further level of addressing for the data, so that it can be directed by the logical unit to the appropriate data stream.

Please refer to the SEND MESSAGE(6) command (see) for a description of the fields in this command.

7.2.6 SEND MESSAGE(12) command

The SEND MESSAGE(12) command (see table 72) transfers data from the application client to the device server.

Table 72 - SEND MESSAGE(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (AAh)							
1	Reserved							
2	Reserved							
3	Reserved							
4	(MSB)			Stream				
5				selection				(LSB)
6	(MSB)							
7				Transfer				
8				length				
9								(LSB)
10	Reserved							
11	Control							

A reservation conflict shall occur when a SEND MESSAGE(12) command is received from an initiator other than the one holding a logical unit reservation.

The stream selection field specifies a further level of addressing for the data, so that it can be directed by the logical unit to the appropriate data stream.

Please refer to the SEND MESSAGE(6) command (see for a description of the fields in this command.

7.3 Parameters for communication devices

7.3.1 Diagnostic parameters

This subclause defines the descriptors and pages for diagnostic parameters used with communication devices.

The diagnostic page codes for communication devices are defined in table 73.

Table 73 - Diagnostic page codes

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

7.3.2 Log parameters

This subclause defines the descriptors and pages for log parameters used with communication devices.

The log page codes for communication devices are defined in table 74.

Table 74 - Log page codes

Page Code	Description	Subclause
01h	Buffer over-run/under-run page	SPC
07h	Last *n* error events page	SPC
06h	Non-media error page	SPC
00h	Supported log pages	SPC
02h - 05h	Reserved	
08h - 2Fh	Reserved	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

7.3.3 Mode parameters

This subclause defines the descriptors and pages for mode parameters used with communication devices.

The mode parameter list, including the mode parameter header and mode block descriptor, are defined in SPC.

The medium-type code field is contained in the mode parameter header (see SPC). This field is reserved for communications devices.

The device specific parameter field is contained in the mode parameter header (see SPC). This field is reserved for communications devices.

The density code field is contained in the mode parameter block descriptor (see SPC). This field is reserved for communications devices.

The mode page codes for communications devices are shown in table 75.

Table 75 - Mode page codes

Page Code	Description	Subclause
0Ah	Control mode page	SPC
02h	Disconnect-reconnect page	SPC
09h	Peripheral device page	SPC
01h	Reserved	
03h - 08h	Reserved	
0Bh - 1Fh	Reserved	
00h	Vendor-specific (does not require page format)	
20h - 3Eh	Vendor-specific (page format required)	
3Fh	Return all pages (valid only for the MODE SENSE command)	

7.4 Definitions specific to communications devices

7.4.1. **communications device:** An SCSI device whose principal function is to communicate with one or more other systems, usually over distances that exceed the maximum cable length defined for SCSI.

7.4.2. **external medium:** The medium used by the communications device to send or receive information to or from one or more communications devices. The other communications devices may or may not use an SCSI interface.

7.4.3. **external protocol:** The protocol(s) used by the communications device to transfer information over the external medium. The external protocol(s) are not defined by this International Standard.

8 Informative Annex

8.1 Density codes

The addition of the REPORT DENSITY SUPPORT command has removed the requirement that density codes be allocated in the stream command set. While none of the following codes are binding, it is suggested that logical units use the listed code for the referenced density. These codes are specified in or returned by the density code field in the device block descriptor of the mode parameter data.

Table 76 - Historical sequential-access density codes

Code Value	Description							Note
0Eh	Reserved for ECMA							
	Descriptions for magnetic tapes							
	Width		Tracks	Density		Code	Type	Reference	
	mm	(in)		bpmm	(bpi)				
01h	12,7	(0,5)	9	32	(800)	NRZI	R	X3.22-1983	2
02h	12,7	(0,5)	9	63	(1 600)	PE	R	X3.39-1986	2
03h	12,7	(0,5)	9	246	(6 250)	GCR	R	X3.54-1986	2
05h	6,3	(0,25)	4/9	315	(8 000)	GCR	C	X3.136-1986	1
06h	12,7	(0,5)	9	126	(3 200)	PE	R	X3.157-1987	2
07h	6,3	(0,25)	4	252	(6 400)	IMFM	C	X3.116-1986	1
08h	3,81	(0,15)	4	315	(8 000)	GCR	CS	X3.158-1987	1
09h	12,7	(0,5)	18	1 491	(37 871)	GCR	C	X3B5/87-099	2
0Ah	12,7	(0,5)	22	262	(6 667)	MFM	C	X3B5/86-199	1
0Bh	6,3	(0,25)	4	63	(1 600)	PE	C	X3.56-1986	1
0Ch	12,7	(0,5)	24	500	(12 690)	GCR	C	HI-TC1	1,6
0Dh	12,7	(0,5)	24	999	(25 380)	GCR	C	HI-TC2	1,6
0Fh	6,3	(0,25)	15	394	(10 000)	GCR	C	QIC-120	1,6
10h	6,3	(0,25)	18	394	(10 000)	GCR	C	QIC-150	1,6
11h	6,3	(0,25)	26	630	(16 000)	GCR	C	QIC-320	1,6
12h	6,3	(0,25)	30	2 034	(51 667)	RLL	C	QIC-1350	1,6
13h	3,81	(0,15)	1	2 400	(61 000)	DDS	CS	X3B5/88-185A	5
14h	8,0	(0,315)	1	1 703	(43 245)	RLL	CS	X3.202-1991	5
15h	8,0	(0,315)	1	1 789	(45 434)	RLL	CS	ECMA TC17	5
16h	12,7	(0,5)	48	394	(10 000)	MFM	C	X3.193-1990	1
17h	12,7	(0,5)	48	1 673	(42 500)	MFM	C	X3B5/91-174	1
18h	12,7	(0,5)	112	1 673	(42 500)	MFM	C	X3B5/92-50	1
19h	TBD								
1Ah	TBD								
1Bh	TBD								
1Ch	6,3	(0,25)	34	1 654	(42 000)	MFM	C	QIC-385M	1,6
1Dh	6,3	(0,25)	32	1 512	(38 400)	GCR	C	QIC-410M	1,6
1Eh	6,3	(0,25)	30	1 385	(36 000)	GCR	C	QIC-1000C	1,6
1Fh	6,3	(0,25)	30	2 666	(67 733)	RLL	C	QIC-2100C	1,6
20h	6,3	(0,25)	144	2 666	(67 733)	RLL	C	QIC-6GB(M)	1,6
21h	6,3	(0,25)	144	2 666	(67 733)	RLL	C	QIC-20GB(C)	1,6
22h	6,3	(0,25)	42	1 600	(40 640)	GCR	C	QIC-2GB(C)	?
23h	6,3	(0,25)	38	2 666	(67 733)	RLL	C	QIC-875M	?
24h	3,81	(0,15)	1	2 400	(61 000)	?	CS	DDS-2	5
25h	3,81	(0,15)	1	3 816	(97 000)	?	CS	DDS-3	5
26h	3,81	(0,15)	1	3 816	(97 000)	?	CS	DDS-4	5
27h	8,0	(0,315)	1	3 056	(77 611)	RLL	CS	TBD	5
28h	12,7	(0,5)	36	1 491	(37 871)	GCR	C	X3B5/94-043A	1
29h	12,7	(0,5)							
2Ah									
2Bh	12,7	(0,5)							

Code	Description
NRZI	Non return to zero, change on ones
GCR	Group code recording
PE	Phase encoded
IMFM	Inverted modified frequency modulation
MFM	Modified frequency modulation
DDS	DAT data storage
RLL	Run length limited

Type	Description
R	Reel-to-reel
C	Cartridge
CS	Cassette

NOTES

1 Serial recorded.
2 Parallel recorded.
3 Old format known as QIC-11.
5 Helical scan.
6 This is not an American National Standard. The reference is based on an industry standard definition of the media format.