INTERNATIONAL STANDARD

ISO/IEC 14776-331

First edition
2002-09

Information technology – Small computer system interface (SCSI) –

Part 331: Stream commands (SSC)

Reference number
ISO/IEC 14776-331:2002(E)

INTERNATIONAL STANDARD

ISO/IEC 14776-331

First edition
2002-09

Information technology – Small computer system interface (SCSI) –

Part 331: Stream commands (SSC)

PRICE CODE X

For price, see current catalogue

CONTENTS

FOREWORD 6
INTRODUCTION 7
1 Scope 8
2 Normative references 9
3 Definitions, symbols and abbreviations 9
3.1 Definitions 9
3.2 Symbols and abbreviations 11
3.3 Keywords 11
3.4 Conventions 12
4 General 13
4.1 Overview 13
4.2 Physical models 13
5 Sequential-access devices 13
5.1 Definitions specific to sequential access devices 13
5.2 Sequential-access device model 16
5.2.1 Physical elements 16
5.2.2 Early warning 19
5.2.3 Partitions within a volume 19
5.2.4 Logical elements within a partition 21
5.2.5 Data buffering 22
5.2.6 Tagged command queuing 22
5.2.7 Recorded object descriptors (block identifiers) 23
5.2.8 Direction and position definitions 23
5.2.9 Write protection 25
5.2.10 Progress indication 27
5.2.11 TapeAlert application client interface 28
5.2.12 Device reservations and command behavior 31
5.3 Command descriptions for sequential-access devices 32
5.3.1 ERASE command 34
5.3.2 FORMAT MEDIUM command 34
5.3.3 LOAD UNLOAD command 36
5.3.4 LOCATE command 38
5.3.5 READ command 39
5.3.6 READ BLOCK LIMITS command 41
5.3.7 READ POSITION command 42
5.3.8 READ REVERSE command 45
5.3.9 RECOVER BUFFERED DATA command 46
5.3.10 REPORT DENSITY SUPPORT command 47
5.3.11 REWIND command 50
5.3.12 SPACE command 51
5.3.13 VERIFY command 54
5.3.14 WRITE command 55
5.3.15 WRITE FILEMARKS command 57

5.4 Parameters for sequential-access devices 58
5.4.1 Diagnostic parameters 58
5.4.2 Log parameters 58
5.4.3 Mode parameters 60
6 Printer devices 79
6.1 Model for printer devices 79
6.2 Commands for printer devices 80
6.2.1 FORMAT command 81
6.2.2 PRINT command 82
6.2.3 RECOVER BUFFERED DATA command 83
6.2.4 SLEW AND PRINT command 83
6.2.5 STOP PRINT command 84
6.2.6 SYNCHRONIZE BUFFER command 84
6.3 Parameters for printer devices 85
6.3.1 Diagnostic parameters 85
6.3.2 Log parameters 85
6.3.3 Mode parameters 86
Annex A (informative) Historical density codes 94
Annex B (normative) TapeAlert log page parameter codes (flags) 96
Bibliography 101

Figure 1 – SCSI standards - general structure 8
Figure 2 – Typical volume layout 17
Figure 3 – Typical medium track layout 17
Figure 4 – Serpentine recording example 18
Figure 5 – Parallel recording example 18
Figure 6 – Helical scan recording example 18
Figure 7 – Early-warning example 19
Figure 8 – Partitioning example – one partition per track group 20
Figure 9 – Partitioning example – one partition per two track groups 20
Figure 10 – Partitioning example – two partitions per track group 20
Figure 11 – SCSI printer model 80

Table 1 – Error conditions and sense keys 24
Table 2 – Write protect ASC/ASCQ combinations 26
Table 3 – Commands providing progress indication without changing ready state 27
Table 4 – Commands changing ready state and providing progress indication 28
Table 5 – TapeAlert default informational exceptions control page 29
Table 6 – TapeAlert flag types 30
Table 7 – TapeAlert flags minimum subset 30
Table 8 – TapeAlert flag definitions 31
Table 9 – Streaming commands that are allowed in the presence of various reservations 32
Table 10 – Commands for sequential-access devices 33
Table 11 – ERASE command 34
Table 12 – FORMAT MEDIUM command 35
Table 13 – Format field definition 36

Table 14 – LOAD UNLOAD command 37
Table 15 – LOCATE command 38
Table 16 – READ command 39
Table 17 – READ BLOCK LIMITS command 41
Table 18 – READ BLOCK LIMITS data 41
Table 19 – READ POSITION command 42
Table 20 – READ POSITION data format, short form 43
Table 21 – READ POSITION data format, long form 44
Table 22 – READ REVERSE command 45
Table 23 – RECOVER BUFFERED DATA command 46
Table 24 – REPORT DENSITY SUPPORT command 47
Table 25 – Density support header 48
Table 26 – Density support data block descriptor 48
Table 27 – REWIND command 51
Table 28 – SPACE command 51
Table 29 – Code definition 52
Table 30 – VERIFY command 54
Table 31 – WRITE command 55
Table 32 – WRITE FILEMARKS command 57
Table 33 – Diagnostic page codes 58
Table 34 – Log page codes 59
Table 35 – Parameter codes for sequential-access device page 60
Table 36 – TapeAlert log page 60
Table 37 – Device-specific parameter 61
Table 38 – Buffered modes 61
Table 39 – Speed field definition 61
Table 40 – Sequential-access density codes 62
Table 41 – Mode page codes 63
Table 42 – Data compression page 64
Table 43 – Possible boundaries and resulting sense keys due to data compression 65
Table 44 – Compression algorithm identifiers 67
Table 45 – Device configuration page 67
Table 46 – EOD defined values 69
Table 47 – Medium partition page(1) 71
Table 48 – PSUM values 72
Table 49 – Medium format recognition values 74
Table 50 – Medium partition page(2-4) 75
Table 51 – Read-write error recovery page 76
Table 52 – Informational exceptions control page 78
Table 53 – TapeAlert test descriptions 78
Table 54 – Commands for printer devices 81
Table 55 – FORMAT command 82
Table 56 – Format type values 82

Table 57 – PRINT command 82
Table 58 – RECOVER BUFFERED DATA command 83
Table 59 – SLEW AND PRINT command 83
Table 60 – STOP PRINT command 84
Table 61 – SYNCHRONIZE BUFFER command 85
Table 62 – Diagnostic page codes 85
Table 63 – Log page codes 86
Table 64 – Printer device-specific parameter 86
Table 65 – Mode page codes 87
Table 66 – Parallel printer interface 87
Table 67 – Parity select codes 88
Table 68 – VFU control byte 88
Table 69 – Printer options 89
Table 70 – Font identification values 89
Table 71 – Slew mode codes 90
Table 72 – Line slew codes 90
Table 73 – Form slew codes 91
Table 74 – Data termination option codes 91
Table 75 – Serial printer interface 92
Table 76 – Parity selection codes 92
Table 77 – Pacing protocol codes 93
Table A.1 – Historical sequential-access density codes 94
Table B.1 – TapeAlert log page parameter codes 96

INFORMATION TECHNOLOGY – SMALL COMPUTER SYSTEM INTERFACE (SCSI) –

Part 331: Stream commands (SSC)

FOREWORD

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

2) In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

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

International Standard ISO/IEC 14776-331 was prepared by subcommittee 25: Interconnection of information technology equipment, of ISO/IEC joint technical committee 1: Information technology.

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

INTRODUCTION

This standard specifies functional requirements for SCSI-3 Stream commands (SSC). SSC permits SCSI streaming devices such as tape and printer devices to attach to computers and provides the definitions for their use.

This standard specifies the external behavior of a device server that defines itself as either a Sequential-access device or a Printer device in the device type field of the INQUIRY command response data. Together, these device types are known as Stream Devices. The SSC standard conforms to SCSI-3 Architectural Model (ISO/IEC 14776-411:1999) standard.

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

The SCSI-3 Stream Commands (SSC) standard specifies a protocol for command-level communications between an application client and a device server that has identified itself as a stream device.

The SCSI-3 Stream Commands (SSC) standard encompasses the following:

– Clause 1 describes the scope.
– Clause 2 lists the normative references.
– Clause 3 provides descriptions, symbols and abbreviations used in this standard.
– Clause 4 provides an overview of the stream device class and command set.
– Clause 5 specifies a model (including the TapeAlert interface definition), command set and parameters for sequential-access devices.
– Clause 6 specifies a model, command set and parameters for printer devices.
– Annex A provides the density code list for sequential-access devices.
– Annex B provides a list of TapeAlert log page parameter codes (flags).

INFORMATION TECHNOLOGY – SMALL COMPUTER SYSTEM INTERFACE (SCSI) –

Part 331: Stream commands (SSC)

1 Scope

This part of ISO/IEC 14776 defines the command set extensions to facilitate operation of SCSI stream devices. This standard in conjunction with ANSI INCITS 351-2001 fully specifies the standard command set for the SCSI stream device class.

The objective of this standard (SSC) is to provide the following:

a) permit an application client to communicate over a SCSI service delivery subsystem, with a logical unit that declares itself to be a sequential access device or printer device in the device type field of the INQUIRY command response data;

b) define commands unique to each type of SCSI stream device;

c) define commands to manage the operation of SCSI stream devices; and

d) define the differences between the types of SCSI stream devices.

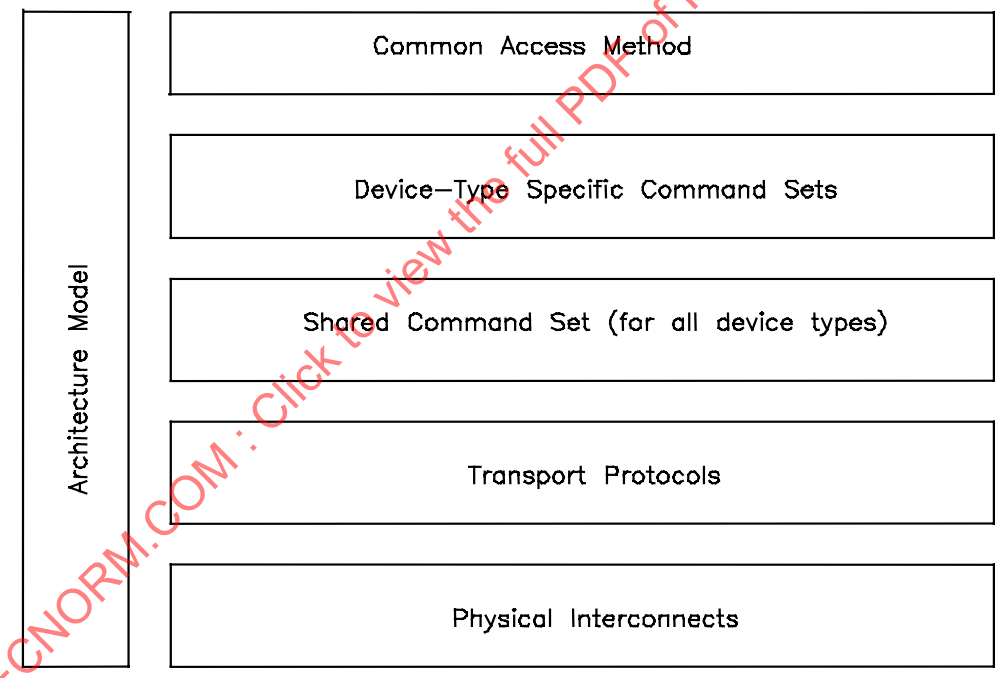

Figure 1 – SCSI standards – General structure

Figure 1 shows the general structure of SCSI standards. The figure is not intended to imply a relationship such as a hierarchy, protocol stack or system architecture.

2 Normative references

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

ISO/IEC 9316:1995, *Information technology – Small Computer System Interface-2*

ISO/IEC 14776-321:2002, *Information technology – Small Computer System Interface-3 (SCSI-3) – Part 321: Block commands (SBC)*

ANSI INCITS 351-2001, *Information technology – SCSI Primary Commands-2 (SPC-2)*[1]

3 Definitions, symbols and abbreviations

The term SCSI is used to refer to the family of standards described in this clause. The Small Computer System Interface – 2 standard (ISO/IEC 9316:1995) and its architecture are referred to herein as SCSI-2.

3.1 Definitions

For the purpose of this document the following definitions apply. The terms cited do not constitute a comprehensive glossary for SCSI.

3.1.1
application client
object that is the source of SCSI commands (see SCSI SAM-2)[2]

3.1.2
auxiliary memory
memory component that is accessible to the device server

This memory is usually non-volatile and independent of the main function of the device server.

3.1.3
byte
an 8-bit construct

3.1.4
command
request describing a unit of work to be performed by a device server (see SCSI SAM-2)

3.1.5
command descriptor block (CDB)
structure used to communicate commands from an application client to a device server

3.1.6
device server
object within a logical unit that executes SCSI tasks according to the rules of task management (see SCSI SAM-2)

---

1 ISO/IEC 14776-312, *Information technology – Small Computer System Interface (SCSI-2) – Part 312: Primary commands-2 (SPC-2)* is under consideration, see also Bibliography.

2 See Bibliography: reference ISO/IEC 14776-412.

3.1.7
field
group of one or more contiguous bits

3.1.8
information field
command-specific field in the sense data (see ANSI INCITS 351-2001)

3.1.9
initiator
SCSI device capable of supporting one or more application clients that originate device service requests to be processed by a target SCSI device (see ANSI INCITS 351-2001)

3.1.10
medium auxiliary memory (MAM)
auxiliary memory residing on a medium, for example a tape cartridge

3.1.11
one
logical true condition of a variable

3.1.12
page
several commands use regular parameter structures referred to as pages. Each page is identified with a value known as a page code

3.1.13
SCSI device
device connected to a service delivery subsystem supporting a SCSI application protocol (see SCSI SAM-2)

3.1.14
sense data
data returned by a REQUEST SENSE command (see ANSI INCITS 351-2001)

3.1.15
sense key
field in the sense data (see ANSI INCITS 351-2001)

3.1.16
target
SCSI device containing one or more logical units and their associated device servers which receive and perform commands from an initiator (see ANSI INCITS 351-2001)

3.1.17
TapeAlert
software application that provides detailed device diagnostic information using a standard interface

3.1.18
zero
logical false condition of a variable

3.2 Symbols and abbreviations

BOM	beginning-of-medium
BOP	beginning-of-partition
CDB	command descriptor block
ECMA	European Computer Manufacturers' Association
EOD	end-of-data
EOM	end-of-medium
EOP	end-of-partition
EW	early-warning
I/O	input/output
ID	identifier
LSB	least significant bit
MSB	most significant bit
NA	not applicable
Rsvd	reserved
SAM-2	SCSI Architecture Model – 2 (SCSI SAM-2)
SBC	SCSI-3 Block Commands (ISO/IEC 14776-321)
SCSI	either SCSI-2 or SCSI-3
SCSI-2	Small Computer System Interface – 2
SCSI-3	Small Computer System Interface – 3
SMC	SCSI-3 Medium Changer Commands (ISO/IEC 14776-351)
SPC-2	SCSI Primary Commands–2 (ANSI INCITS 351-2001)
SSC	SCSI-3 Stream Commands (ISO/IEC 14776-331 - this document)

3.3 Keywords

Several keywords are used to differentiate between different levels of requirements and optionality, as follows.

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

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

3.3.3
may
indicates flexibility of choice with no implied preference (equivalent to "may or may not")

3.3.4
may not
indicates flexibility of choice with no implied preference (equivalent to "may or may not")

3.3.5
shall
indicates a mandatory requirement. Designers are required to implement all such mandatory requirements to ensure interoperability with other standard conformant products

3.3.6
should
indicates flexibility of choice with a strongly preferred alternative. Equivalent to the phrase “it is recommended”

3.3.7
obsolete
indicates items that were defined in prior SCSI standards but have been removed from this standard

3.3.8
mandatory
indicates items required to be implemented as defined by this standard

3.3.9
optional
describes features that are not required to be implemented by this standard. However, if any optional feature defined by the standard is implemented, it shall be implemented as defined by this standard

3.3.10
reserved
refers to bits, bytes, words, fields and code values that are set aside for future standardization. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as an error

3.3.11
vendor-specific
items (for example a bit, field, code value, etc.) that are not defined by this standard and may be vendor defined

3.4 Conventions

Lower case is used for words having the normal English meaning. Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.3 or in the text where they first appear.

Listed items in this standard do not represent any priority. Any priority is explicitly indicated. Formal lists (e.g., a) red; b) blue; c) green) which use letters are in an arbitrary order. Formal lists (e.g., 1) red; 2) blue; 3) green) which use numbers are in a required sequential order.

If a conflict arises between text, tables, or figures, the order of precedence to resolve conflicts is text, then tables, and finally figures. Not all tables or figures are fully described in text. Tables are used to show data format and values.

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

The following additional conventions are used:

a) The names of abbreviations, commands and acronyms used as signal names are in all uppercase (e.g., IDENTIFY DEVICE);

b) field names are in small caps to distinguish them from normal English;

c) fields containing only one bit are referred to as the name bit instead of the name field;

d) numbers, except clause numbers, that are not immediately followed by lowercase b or h are decimal values;

e) numbers immediately followed by lower-case b (xxb) are binary values;

f) numbers and the upper case letters A, B, C, D, E, and F immediately followed by lower-case h (xxh) are hexadecimal values;

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

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

4 General

4.1 Overview

The SCSI stream device class specifies the behavior of a logical unit that is primarily a streaming data device. Two device types are members of this class: sequential-access and printer devices. Both have the common characteristic of primarily handling data in a sequential manner (i.e., a stream). This does not limit the device's ability to position randomly within the data. However, the physical model defined for each of these device types indicates whether random-access operations are impossible or just inefficient.

A sequential-access device is not truly random-access (see ISO/IEC 14776-321 for a description of a random-access device). Commands to read and write on a sequential-access device do not contain any positioning information fields. Instead, the device position is normally determined by previous commands. Commands are available for absolute and relative positioning. Writing to a sequential-access device may cause all data starting at the point at which the data is written to be invalidated. There may be restrictions on where write operations may be initiated. Reading or writing data as a long string of data, as in a stream, tends to be the most efficient.

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

4.2 Physical models

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

In the printer model, the read/write mechanism may only write and the medium may only move from the supply. Unlike a sequential-access device, data may be formatted on a printer device by font selection and positioning options.

5 Sequential-access devices

5.1 Definitions specific to sequential access devices

5.1.1
beginning-of-medium
extreme position along the medium in the direction away from the supply reel that is accessible by the device. This position may not coincide with a beginning-of-partition position

5.1.2
beginning-of-partition
position at the beginning of the permissible recording region of a partition

If only one partition is defined for a volume, this position is typically equivalent to the beginning-of-medium. This position may not coincide with a beginning-of-medium position.

5.1.3
buffered mode
mode of data transfer in write operations which facilitates tape streaming (see 5.2.5), as reported in the mode parameter header device-specific parameter (see 5.4.3). Buffered mode is indicated by a non-zero value (1h or 2h) in the buffer mode field in the mode parameter header (see 5.4.3). Buffered mode is the opposite of unbuffered mode (see 5.1.21)

5.1.4
early-warning
physical mark or device computed position near but logically before the end-of-partition, independent of physical direction (see 5.2.2)

5.1.5
end-of-data
recorded indication that no valid logical elements are recorded between this position and end-of-partition

End of data is denoted in a format-specific manner (see 5.2.4).

5.1.6
end-of-medium
extreme position along the medium in the direction away from the take-up reel that is accessible by the device

This position may not coincide with an end-of-partition position. This position may be accessed by logical units that support the LOAD UNLOAD command with the eot bit set to one (see 5.3.3).

5.1.7
end-of-partition
position at the end of the permissible recording region of a partition

5.1.8
filemark
special recorded element within a partition, not containing user data, which provides a segmentation scheme for the contents of a partition

5.1.9
gap
non-data element recorded on the medium

Gaps may be recorded between logical elements. The format and method of recording a gap may vary.

5.1.10
logical block
logical element that is a unit of data supplied or requested by an initiator

5.1.11
logical element
unit of data, either a block or a mark

Each logical element has a unique logical block identifier (see 5.2.7), if supported, within the partition.

5.1.12
mark
logical element that does not contain any initiator defined data

A mark is either a setmark or filemark.

5.1.13
overlength
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 or the mode header block size field, whichever is appropriate

5.1.14
partition
entire usable region for recording and reading in a volume or in a portion of a volume, defined in a vendor-specific manner

5.1.15
principal density code
principal density code is a density code selected by the device server

The logical unit indicates the principal density code by reporting a default bit of one in the density support data block descriptor for supported densities in response to the REPORT DENSITY SUPPORT command (see 5.3.10). The selection of the principal density code is vendor-specific.

5.1.16
setmark
special recorded element within a partition, not containing user data, that provides a segmentation scheme similar to filemarks

Setmarks may be ignored based on the rsmk mode parameter (see 5.4.3.2).

5.1.17
spacing
act of positioning the medium on a sequential access device

5.1.18
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 reels, possibly within a cassette or cartridge.

5.1.19
track
contiguous line on the medium consisting of a pattern of recorded signals written by one write component

5.1.20
track group
set of tracks that are recorded at the same time

5.1.21
unbuffered mode
mode of operation where write data is written directly to the medium without being buffered

Unbuffered mode is indicated by a zero value (0h) in the buffer mode field in the mode parameter header (see 5.4.3). Unbuffered mode is the opposite of buffered mode (see 5.1.3).

5.1.22
underlength
incorrect length condition that exists after executing a read command when the requested transfer length in the command descriptor block or the mode header block size field, whichever is appropriate, exceeds the length of the actual block read

5.1.23
volume
recording medium together with its physical carrier

5.2 Sequential-access device model

5.2.1 Physical elements

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

Sequential-access devices are described herein 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 International 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 volume is composed of the recording medium and its physical carrier (for example, reel, cartridge, cassette). Volumes have an attribute of being mounted or de-mounted on a suitable transport mechanism.

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

The RESERVE and RELEASE commands (see ANSI INCITS 351-2001) are mandatory while the PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands (see ANSI INCITS 351-2001) are optional for sequential-access devices. Element 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 (for example, mechanical lock). Other mechanisms for write protect, including software controlled methods, may be available (see 5.2.9). When a logical unit is in write protected state, writing to a medium is prohibited. Any of the following commands could end with CHECK CONDITION status because write protection is enabled: COPY, COPY AND VERIFY, ERASE, FORMAT MEDIUM, MODE SELECT, WRITE, and WRITE FILEMARKS. In this situation, the sense key is DATA PROTECT.

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

As shown in Figure 2, a portion of the physical length of a medium is not usable for recording data. For most volumes, a length of the medium is reserved between the take-up reel and the beginning-of-medium, and between the end-of-medium position and the supply reel. This is done to provide a sufficient tape wrap onto the reel hub and to ensure that recording starts in an undamaged section of the medium.

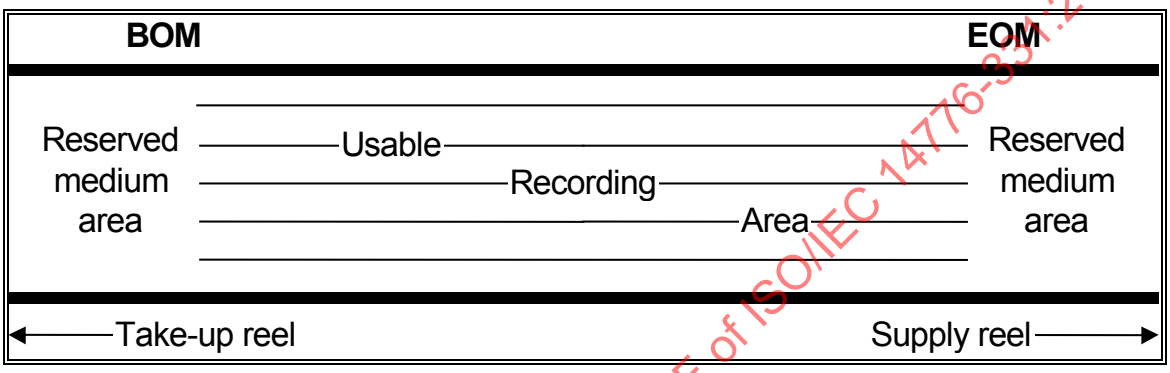

Figure 2 – Typical volume layout

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

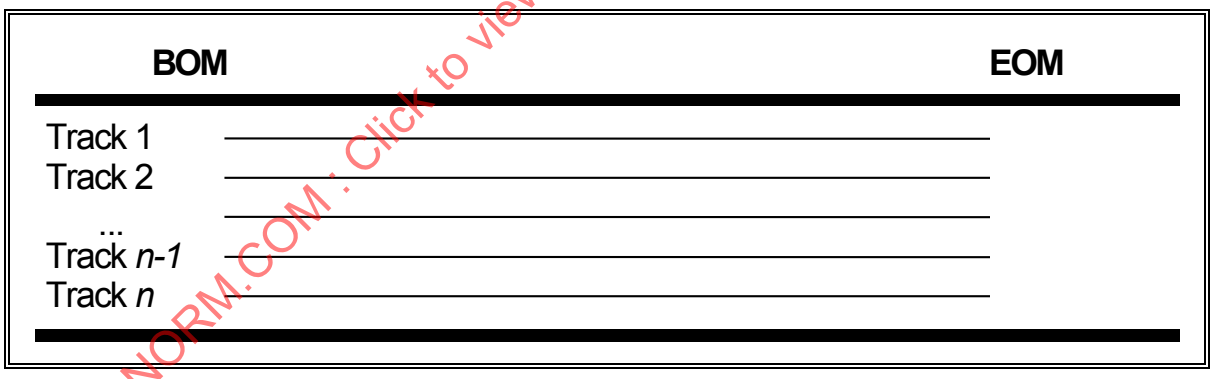

Figure 3 – Typical medium track layout

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

In serpentine recording, not all tracks are recorded at the same time. At the end-of-medium or beginning-of-medium, the device reverses direction and begins recording the next track group. The process of reversing direction and recording the next track group may be repeated until all track groups are recorded. For serpentine devices that record only one track at a time, each physical track represents one track group (see Figure 4).

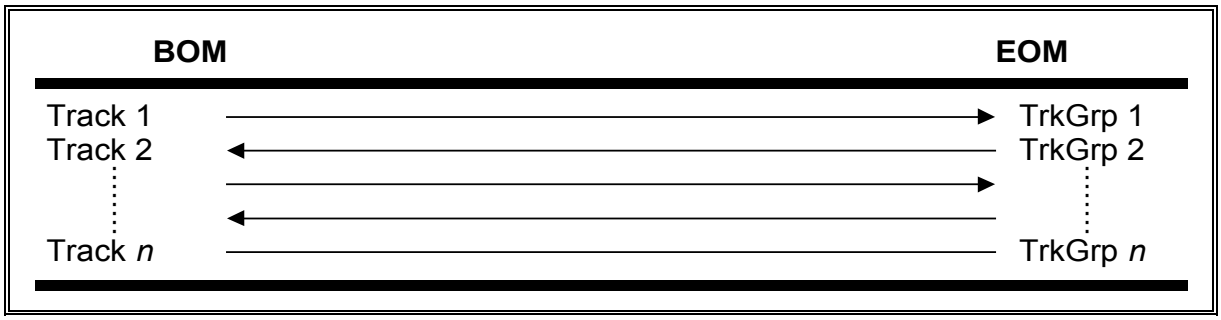

Figure 4 – Serpentine recording example

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).

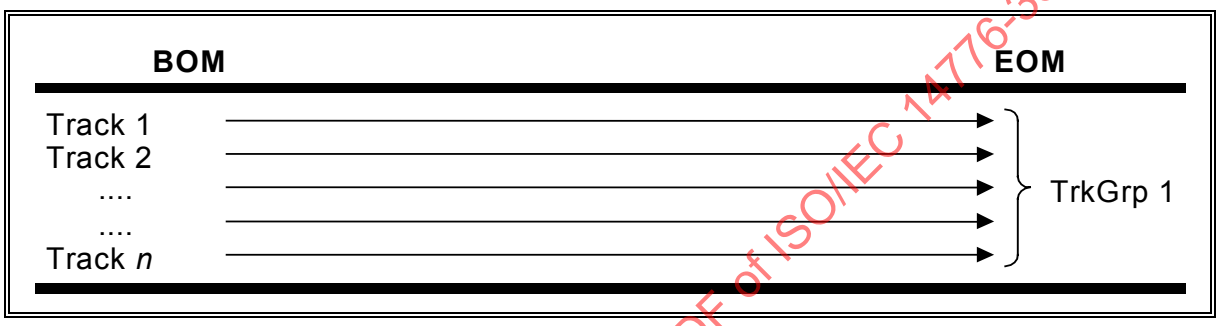

Figure 5 – Parallel recording example

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. One or more tracks may be recorded at the same time. This recording technique is known as helical scan (see Figure 6).

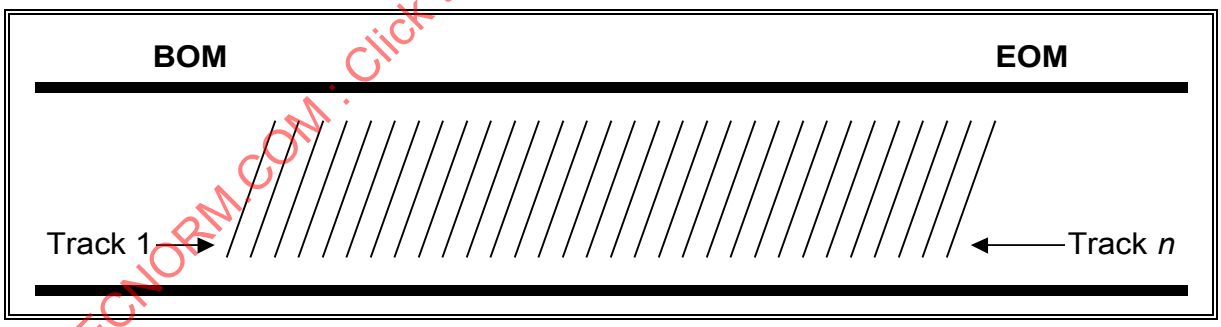

Figure 6 – Helical scan recording example

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

5.2.2 Early warning

When writing, the application client needs an indication that it is approaching the end of the permissible recording area when moving in a direction toward the end of the partition. This position, called early-warning (EW), is typically reported to the application client at a position early enough for the device to write any buffered data to the medium while still leaving enough room for additional recorded data or filemarks (see Figure 7). Some International Standards include physical requirements for a marker placed on the medium to be detected by the device as early-warning.

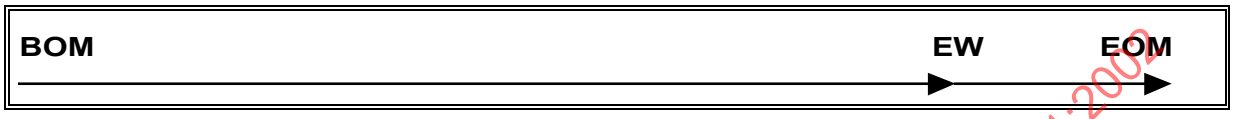

Figure 7 – Early-warning example

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

5.2.3 Partitions within a volume

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

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

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

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

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

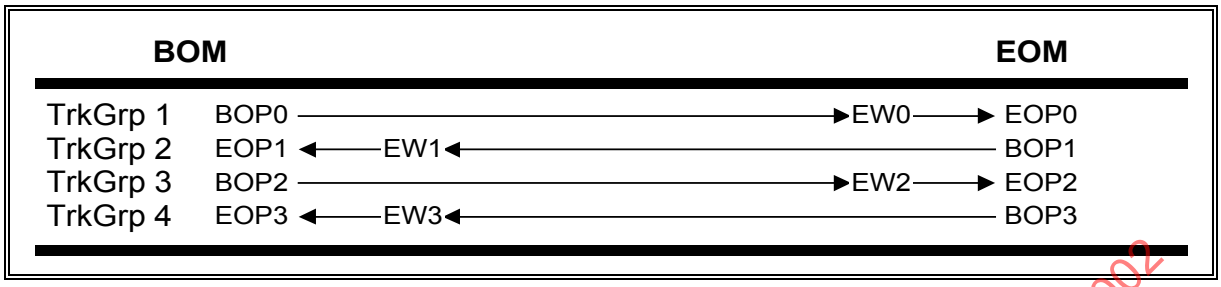

Figure 8 – Partitioning example – One partition per track group

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

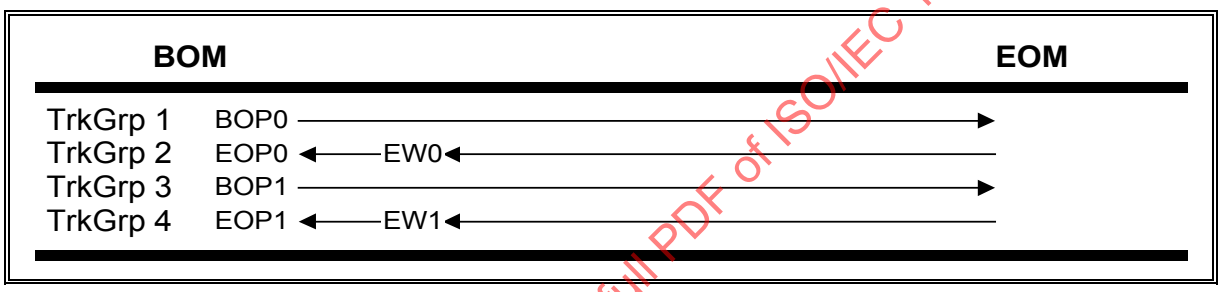

Figure 9 – Partitioning example – One partition per two track groups

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.

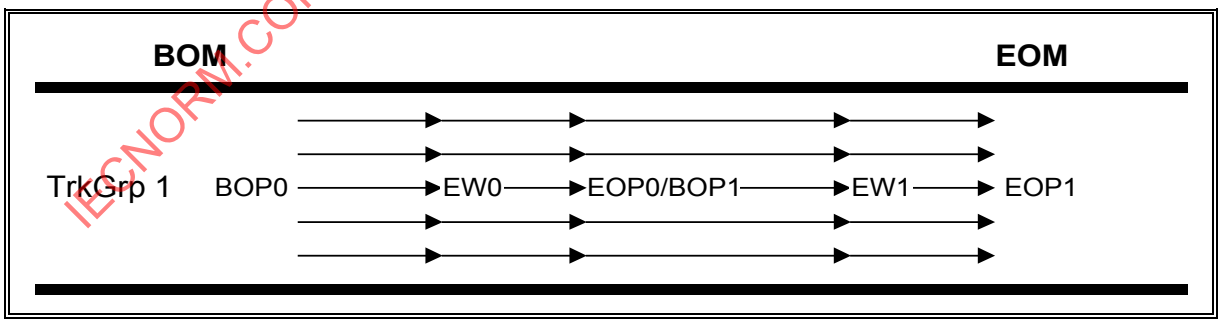

Figure 10 – Partitioning example – Two partitions per track group

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

a) device-defined fixed locations;

b) device-defined based on an application client supplied number of partitions and a vendor-specific allocation algorithm; and

c) definition by partition number and capacity by an application client.

5.2.4 Logical elements within a partition

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

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

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

Setmarks are optional special recorded elements not containing user data. A setmark is another type of special recorded element that does not contain user data, providing an additional method of segmenting data. 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. Unlike filemarks, setmarks may be ignored when reading data or spacing. If ignored, setmarks are skipped when encountered. If not ignored, setmarks terminate reading of data and spacing data or filemarks. If implemented, 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 application client. Minimum and maximum lengths for inter-block gaps are defined by the recording format. In some devices, the length of inter-block gaps may be selected by the application client, using the MODE SELECT command.

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 by the recording format.

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

A sequential-access device may be capable of supporting fixed 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 recording format. Refer to the READ BLOCK LIMITS command (see 5.3.6) for additional information about fixed and variable block mode.

5.2.5 Data buffering

A device may contain a temporary storage area capable of holding one or more logical blocks – i.e. 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 that have not been written to the medium. A synchronize operation has no effect on a data buffer that contains only read-ahead data or write data that has already been successfully written to the medium.

If an unrecoverable write error occurs while in buffered mode, the device generates an error condition to the current active command. If no command is active, the error may be reported on the next applicable operation as a deferred error (see ANSI INCITS 351-2001). For some implementations, asynchronous event reporting or auto contingent allegiance may be required. Refer to SCSI SAM-2 for descriptions of asynchronous event reporting and auto 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 that encounters an unrecoverable error during a read-ahead operation shall not report the error unless 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 10. The WRITE BUFFER command shall ensure transfer of buffered data for modes 4 through 7 (download microcode operations) before performing the download operation. 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.

5.2.6 Tagged command queuing

A device may optionally choose to implement support for tagged command queuing. Provided the initiator does not limit the number of outstanding tagged commands, issuing tagged write commands with data buffering disabled provides the functional equivalent of issuing write commands with data buffering enabled.

5.2.7 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 the medium. When not specified by the format standard, the block identifier value shall be a sequentially increasing number assigned to each logical block, filemark, and setmark recorded in the partition starting with zero for the recorded element at BOP.

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

Some recording 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 recording 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 recording format dependent identifiers. For some implementations, this value may correspond to a 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 application client 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 a 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 and is defined by the recording format specification.

5.2.8 Direction and position definitions

5.2.8.0 General

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

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

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

The concept of being at some position 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) that 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 that 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.

5.2.8.1 Error reporting

If any of the following conditions occur during the execution of a command or if a deferred error prevented the command from executing, the device server shall return CHECK CONDITION status. The appropriate sense key, additional sense code and additional sense code qualifier should be set. The following list (see Table 1) 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.

Table 1 – Error conditions and sense keys

Condition	Sense key
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
Overlength 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, format, partition, or write-type operation on write protected medium	DATA PROTECT
Deferred write error	MEDIUM ERROR VOLUME OVERFLOW HARDWARE ERROR

Refer to the READ command (see 5.3.5) for a description of the fixed bit. Refer to ANSI INCITS 351-2001 for a description of the sense data valid bit and information field contained in the REQUEST SENSE sense data.

The read-write error recovery page (see 5.4.3.5) current values specify behavior when an unrecoverable read or write error is encountered. If this page is not implemented, the behavior is vendor-specific.

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

In the case of an unrecovered write error, if unbuffered mode is selected and the fixed bit is set to one, the sense data valid bit shall be set to one and the information field shall be set to the requested transfer length minus the actual number of 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 sense data 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 initiator for this command 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 initiator for this command 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 reporting. If buffered mode 2h is selected, the error shall be reported to the initiator with unwritten data in the buffer.

In the case of a write attempt to a write protected medium, the additional sense information indicates the cause of the DATA PROTECT sense key (see 5.2.9).

5.2.9 Write protection

5.2.9.0 General

Write protection of the volume prevents the changing of logical data on the volume by commands issued to the device server. Write protection may result from hardware controls (such as tabs on the medium housing), conditions such as positioning within unrecoverable data, or software write protects. All of these sources of write protects are independent. When present, any of these conditions shall cause otherwise valid commands that request alteration of the format or data on the volume to be rejected with a CHECK CONDITION status with the sense key set to DATA PROTECT (see 5.2.9.1). Only if all of the write protects are disabled shall the device server accept commands that require writing to the medium to complete.

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

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

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

a) software write protect for the device server across mounts;

b) associated write protect for the currently mounted volume;

c) persistent write protect of a volume across mounts; and

d) permanent write protect of a volume across mounts.

The application client may control these write protects using the MODE SELECT command with the control mode page (see ANSI INCITS 351-2001) and the device configuration page (see 5.4.3.2). All of the software write protection methods are optional. Changing the state of any software write protect shall not prevent previously buffered write data from transferring to the medium.

5.2.9.1 Write protect additional sense code and additional sense code qualifier use

The additional sense code and additional sense code qualifier (ASC/ASCQ) combination associated with the DATA PROTECT sense key depends on the write protection in effect at the time. Table 2 specifies the preferred ASC/ASCQ combination for the given write protection. Alternatively, the generic ASC/ASCQ of WRITE PROTECTED may be returned by the device server.

Table 2 – Write protect ASC/ASCQ combinations

Cause of DATA PROTECT error	ASC/ASCQ
Hardware Write Protect	HARDWARE WRITE PROTECTED
Permanent Write Protect	PERMANENT WRITE PROTECT
Persistent Write Protect	PERSISTENT WRITE PROTECT
Associated Write Protect	ASSOCIATED WRITE PROTECT
Software Write Protect	LOGICAL UNIT SOFTWARE WRITE PROTECTED

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

5.2.9.2 Software Write Protect for the device server

Software Write Protect for the device server controls write protection for the device server. This method of write protect is optionally controlled from the control mode page (see ANSI INCITS 351-2001) or the swp bit in the device configuration page (see 5.4.3.2). Either or both methods may be implemented by the device server. If both methods are implemented, each control bit is independently set. Software write protection exists if either bit is non-zero. The state of software write protect for the device server shall not be recorded on the medium. The value of the swp bit may be altered by the application client (if the swp bit is changeable). The state of each control bit shall be reset to its default state on a reset or power-up condition.

5.2.9.3 Associated Write Protect

Associated Write Protect controls write protection for the currently mounted volume as long as the current volume is mounted. The associated write protect state is controlled by the asocwp bit in the device configuration page (see 5.4.3.2). Associated write protection exists if the asocwp bit is non-zero. Associated write protection may be altered by the application client (if the asocwp bit is changeable) if a volume is mounted. If a volume is de-mounted or if a reset or power-up condition occurs, associated write protection shall be removed.

5.2.9.4 Persistent Write Protect

Persistent Write Protect controls write protection for the currently mounted volume. The persistent write protect state is controlled by the perswp bit in the device configuration page (see 5.4.3.2). If enabled, persistent write protection shall exist for the mounted volume until disabled by the application client. The state of persistent write protection shall be recorded with the volume. The device server shall report the perswp bit as one when a mounted volume is marked with persistent write protection. If a volume is de-mounted or if a reset or power-up condition occurs, the device server shall report the perswp bit as zero prior to the mounting of a volume. The means for recording the state of persistent write protect for the volume may be specified in the applicable recording format standard or be vendor-specific.

5.2.9.5 Permanent Write Protect

Permanent Write Protect controls write protection for the currently mounted volume. The permanent write protect state is controlled by the prmwp bit in the device configuration page (see 5.4.3.2). If enabled, permanent write protection shall exist for the mounted volume until disabled by a vendor-specific method. The state of permanent write protection shall be recorded with the volume. The device server shall report the prmwp bit as one when a mounted volume is marked with permanent write protection. If a volume is de-mounted or if a reset or power-up condition occurs, the device server shall report the prmwp bit as zero prior to the mounting of a volume. The means for recording the state of permanent write protect for the volume may be specified in the applicable recording format standard or be vendor-specific.

Permanent write protection shall not be removed by a MODE SELECT command using the prmwp bit. Methods to remove this protection may or may not exist and are vendor-specific.

5.2.10 Progress indication

For the following immediate operations where the device server remains ready, an application client may follow the progress of the operation (see Table 3).

Table 3 – Commands providing progress indication without changing ready state

Operation	Options	Subclause
ERASE	immed = 1, long = 1	5.3.1
LOCATE	immed = 1	5.3.4
REWIND	immed = 1	5.3.11
VERIFY	immed = 1	5.3.13

If the immed bit is one, an initiator not subject to a reservation conflict may receive a deferred error indication on any subsequent command. While the device server is performing the immediate operation, an application client may test the progress of the operation by interpreting the progress indication information in the sense-key specific field of the sense data. During the operation, the device server shall report a sense key value of NO SENSE and additional sense information of OPERATION IN PROGRESS. The device server should use the sense key specific function for progress indication to provide information on the completion of the operation.

For the following immediate operations where the device server is ready or will become ready, an application client may follow the progress of the operation (see Table 4).

Table 4 – Commands changing ready state and providing progress indication

Operation	Options	Subclause
FORMAT MEDIUM	immed = 1	5.3.2
LOAD UNLOAD	immed = 1, load = 1, eot = 0	5.3.3
LOAD UNLOAD	immed = 1, load = 0, eot = 1,	5.3.3

If the immed bit is one, an initiator not subject to a reservation conflict may receive a deferred error indication on any subsequent command. While the device server is performing the immediate operation, an application client may test the progress of the operation by interpreting the progress indication information in the sense-key specific field of the sense data. During the operation, the device server shall report a sense key value of NOT READY and additional sense information of LOGICAL UNIT NOT READY, OPERATION IN PROGRESS, NOT READY, FORMAT IN PROGRESS or LOGICAL UNIT IS IN PROCESS OF BECOMING READY, as appropriate. The sense key specific function for progress indication may be used by the device server to provide information on the completion of the operation.

NOTE 1 A REQUEST SENSE command following a TEST UNIT READY command that results in CHECK CONDITION status may provide information, which if acted upon, may lead to unexpected conditions. For example, progress indication reporting is useful when a medium changer is used to service a sequential-access device following an unload operation with IMMED=1b. A TEST UNIT READY command may receive CHECK CONDITION status and a NOT READY sense key reported in the subsequent sense data, which might imply that the unload operation is finished. If the initiator ignores the progress indication information in the sense data, an EXCHANGE MEDIUM or MOVE MEDIUM command (see ISO/IEC 14776-351) to move the dismounted volume from the device may fail to grab the volume if the unload operation is still in progress.

5.2.11 TapeAlert application client interface

5.2.11.0 General

TapeAlert information shall be accessed using LOG SENSE page 2Eh (see Table 36). TapeAlert configuration shall be performed using MODE SELECT page 1Ch (see Table 52). The MODE SENSE and MODE SELECT configuration of the TapeAlert interface is compatible with the informational exceptions control page. The application client shall first check the tape drive to determine whether it supports the TapeAlert log page (2Eh). The default application client access to the TapeAlert log page shall be performed using a polling method, with the page control bits in the LOG SENSE command set to 00h.

At minimum, the TapeAlert log page shall be read from the tape drive/autoloader device for the following:

a) at the beginning of a write/read job, even if a medium is not loaded;

b) immediately after an unrecoverable error during the write/read job;

c) at the end of each tape when the write/read job spans multiple tapes. If the tape is to be ejected then the log page shall be read before the tape cartridge is ejected; or

d) at the end of a write/read job.

The application client may also poll the TapeAlert log page at regular intervals (for example, every 60 s) while the tape drive is idle. The application client may use the TapeAlert mode page (1Ch) to configure other access methods, depending on what options are supported by the tape drive.

Each time the application client reads the TapeAlert log page, it shall check all 64 flags (see Table 79) to discover which are set (there may be more than one). The definitions of the 64 flags are device type specific. There is one definition for tape drive/autoloader devices, and a different definition for stand-alone changer devices (in libraries). For each flag set, the application client shall communicate the defined error message and severity for that flag to the user and log it. If multiple flags are set simultaneously, they shall be displayed together in ascending order of severity. At the beginning of each set of TapeAlert error messages, the tape device/library that initiated them shall be identified. For the tape drive medium-related flags (flags 4h, 7h, and Fh), the software label of the medium shall be included in the TapeAlert error messages so that the user is aware what piece of medium the error refers to. Such information may also be displayed with the messages for other flags as well if required. The information read in the TapeAlert flags shall not in itself cause the application client to stop a current backup or restore operation.

5.2.11.1 TapeAlert informational exceptions control page implementation

See Table 52 for a description of the TapeAlert informational exceptions control page. A device may choose not to support any of the changeable TapeAlert mode page fields. The recommended TapeAlert default mode page implementation is described in Table 5.

Table 5 – TapeAlert default informational exceptions control page

Parameter	Description
DExcpt	When set to zero this enables TapeAlert reporting using the specified mrie mode. By default, this means supporting mrie mode 03h that generates a CHECK CONDITION with a RECOVERED ERROR sense key (unless the device has been configured to not allow reporting of recovered errors).
test	When set to one this provides the ability to generate false TapeAlert events, and also set/clear test TapeAlert flags using test flag number.
test flag number	When set to non-zero and test set to one, this provides an interface to specify which TapeAlert flags to set/clear. Note that supporting test flag number does not mean that report count is supported.

5.2.11.2 TapeAlert log sense format

The TapeAlert interface to the tape drive is based on a LOG SENSE page (2Eh) containing 64 one-byte flags (see Table B.1). The specific conditions for any one flag to be set and cleared are vendor-specific. The minimum subset of flags and basic implementation guidelines are described in Table 7.

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

Each flag shall be cleared in the following circumstances:

a) at drive power on;

b) after the TapeAlert log page is read - note in multi-initiator environments the TapeAlert flags shall be cleared on a per-initiator basis such that set flags are still visible to other initiators;

c) when the specified corrective action has been taken (such as using a cleaning cartridge);

d) on SCSI bus reset or bus device reset message; or

e) on LOG SELECT reset.

NOTE 2 The recommended action upon receiving LOG SELECT for the TapeAlert log page is to reject the command with CHECK CONDITION status and set the sense key to ILLEGAL REQUEST.

NOTE 3 When a flag is cleared by reading the TapeAlert log page, a flag cannot be set again until the error condition is removed (i.e. the specified corrective action has been taken). For example, if the cartridge in the drive is not data grade, once flag 8 has been cleared by the application client reading the log page, it cannot be set again until the cartridge has been removed. All other methods of clearing allow the flag to be set again.

There are three types of flags listed in Table 6 in order of increasing severity:

Table 6 – TapeAlert flag types

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

5.2.11.3 Tape drive/autoloader flag definitions

Flag definitions are defined in Table B.1 for the following device type:

a) Tape drive/autoloader (streaming device using a single physical ID). If the device includes an integrated changer device on another LUN under the same physical device ID (for example, an autoloader), then it shall still be treated as a single streaming device.

The minimum subset of flags that shall be supported by a tape drive/autoloader and some basic implementation guidelines are described in Table 7.

NOTE 4 These are only examples and may not relate to some tape technologies.

Table 7 – TapeAlert flags minimum subset

Flag number	Flag type	Implementation guidelines
3h	Hard error	Set for any unrecoverable read/write/positioning error, and is internally cleared when the medium is ejected (this flag is set as in 5, or 6)
4h	Medium	Set for any unrecoverable read/write/positioning error that is due to a faulty medium, and is internally cleared when the medium is ejected.
5h	Read failure	Set for any unrecoverable read error where the diagnosis is uncertain and could either be a faulty medium or faulty drive hardware, and is internally cleared when the medium is ejected.
6h	Write failure	Set for any unrecoverable write/positioning error where the diagnosis is uncertain and could either be a faulty medium or faulty drive hardware, and is internally cleared when the medium is ejected.
14h	Clean now (if drive supports cleaning cartridges)	Set when the tape drive detects it needs cleaning (i.e. when the cleaning LED is lit), and is internally cleared when the drive is successfully cleaned.
16h	Expired cleaning	Set when the tape drive detects a cleaning cycle was attempted but was not successful, and is internally cleared when the next cleaning cycle is attempted.
1Fh	Hardware B	Set when the tape drive fails its internal Power-On-Self-Tests (POST), and is not internally cleared until the drive is powered off.

If the tape drive/autoloader does not support any of the remaining error flags, then it is acceptable for unsupported flags to remain unset.

The tape drive/autoloader flag definitions are grouped into the following sections:

Table 8 – TapeAlert flag definitions

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

5.2.12 Device reservations and command behavior

Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation. Refer to ANSI INCITS 351-2001 for a description of device reservations. The details of which commands are allowed under what types of reservations are described in Table 9. For the reservation restrictions placed on commands for the Reserve/Release management method, see Table 9, column A. For the reservation restrictions placed on commands for the Persistent Reservations management method, see Table 9, columns under B.

If any element is reserved within a logical unit, that logical unit shall be considered reserved for the commands listed in Table 9. The allowed/conflict information in the table shall apply.

In Table 9 the following key words are used:

Allowed: Commands issued by initiators not holding the reservation or by initiators not registered when a registrants only persistent reservation is present completes normally.

Conflict: Commands issued by initiators not holding the reservation or by initiators not registered when a registrants only persistent reservation is present, shall not be performed and the device server shall terminate the command with a RESERVATION CONFLICT status.

N/a: Not applicable.

Commands from initiators holding a reservation should complete normally. The behavior of commands from registered initiators when a registrants only persistent reservation is active is specified in Table 9.

A command that does not explicitly write to the medium shall be checked for reservation conflicts before the command enters the current task state for the first time. Once the command has entered the current task state, it shall not be terminated with a RESERVATION CONFLICT due to a subsequent reservation.

A command that explicitly writes the medium shall be checked for reservation conflicts before the device server modifies the medium or cache as a result of the command. Once the command has modified the medium, it shall not be terminated with a RESERVATION CONFLICT due to a subsequent reservation.

For each command, this standard and the ANSI INCITS 351-2001 standard define the conditions that result in RESERVATION CONFLICT.

Table 9 – Streaming commands that are allowed in the presence of various reservations

Command	Addressed LU is reserved by another Initiator (A)	Addressed LU has this type of persistent reservation held by another Initiator (B)				
		From any Initiator		From Registered Initiator (All RO types)	From Initiator Not Registered	
		Write Excl	Excl Access		Write Excl RO	Excl Access RO
ERASE	Conflict	N/a	Conflict	Allowed	N/a	Conflict
FORMAT MEDIUM	Conflict	N/a	Conflict	Allowed	N/a	Conflict
LOAD UNLOAD	Conflict	N/a	Conflict	Allowed	N/a	Conflict
LOCATE	Conflict	N/a	Conflict	Allowed	N/a	Conflict
READ	Conflict	N/a	Conflict	Allowed	N/a	Conflict
READ BLOCK LIMITS	Allowed	N/a	Allowed	Allowed	N/a	Allowed
READ POSITION	Conflict	N/a	Conflict	Allowed	N/a	Conflict
READ REVERSE	Conflict	N/a	Conflict	Allowed	N/a	Conflict
RECOVER BUFFERED DATA	Conflict	N/a	Conflict	Allowed	N/a	Conflict
REPORT DENSITY SUPPORTED	Allowed	N/a	Allowed	Allowed	N/a	Allowed
REWIND	Conflict	N/a	Conflict	Allowed	N/a	Conflict
SPACE	Conflict	N/a	Conflict	Allowed	N/a	Conflict
VERIFY	Conflict	N/a	Conflict	Allowed	N/a	Conflict
WRITE	Conflict	N/a	Conflict	Allowed	N/a	Conflict
WRITE FILEMARKS	Conflict	N/a	Conflict	Allowed	N/a	Conflict

Key:
LU = Logical Unit;
Excl = Exclusive;
RO = Registrants Only

5.3 Command descriptions for sequential-access devices

5.3.0 General

The commands for sequential-access devices shall be as shown in Table 10. The Flush column indicates whether the command requires buffered data, filemarks, and setmarks to be transferred to the medium.

Refer to Table 9 for a description of device reservations and command behavior.

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

Table 10 – Commands for sequential-access devices

Command name	Flush write data	Operation code	Type	Reference
Obsolete		40h		3.3.7
COMPARE	Yes	39h	O	SPC-2[b]
COPY	Yes	18h	O	SPC-2[b]
COPY AND VERIFY	Yes	3Ah	O	SPC-2[b]
ERASE	Yes	19h	M	5.3.1
FORMAT MEDIUM	No	04h	O	5.3.2
INQUIRY	No	12h	M	SPC-2[b]
LOAD UNLOAD	Yes	1Bh	O	5.3.3
LOCATE	Yes	2Bh	O	5.3.4
LOG SELECT	No	4Ch	O	SPC-2[b]
LOG SENSE	No	4Dh	O	SPC-2[b]
MODE SELECT(6)	Yes[a]	15h	M	SPC-2[b]
MODE SELECT(10)	Yes[a]	55h	O	SPC-2[b]
MODE SENSE(6)	No	1Ah	M	SPC-2[b]
MODE SENSE(10)	No	5Ah	O	SPC-2[b]
MOVE MEDIUM	Yes	A5h	O	SMC[c]
MOVE MEDIUM ATTACHED	Yes	A7h	O	SMC[c]
PERSISTENT RESERVE IN	No	5Eh	O	SPC-2[b]
PERSISTENT RESERVE OUT	No	5Fh	O	SPC-2[b]
PREVENT ALLOW MEDIA REMOVAL	No	1Eh	O	SPC-2[b]
READ	No	08h	M	5.3.5
READ BLOCK LIMITS	No	05h	M	5.3.6
READ BUFFER	Yes	3Ch	O	SPC-2[b]
READ ELEMENT STATUS	No	B8h	O	SMC[c]
READ ELEMENT STATUS ATTACHED	No	B4h	O	SMC[c]
READ POSITION	No	34h	M	5.3.7
READ REVERSE	Yes	0Fh	O	5.3.8
RECEIVE DIAGNOSTIC RESULTS	No	1Ch	O	SPC-2[b]
RECOVER BUFFERED DATA	May	14h	O	5.3.9
RELEASE(6)	No	17h	M	SPC-2[b]
RELEASE(10)	No	57h	M	SPC-2[b]
REPORT DENSITY SUPPORT	No	44h	M	5.3.10
REPORT LUNS	No	A0h	M	SPC-2[b]
REQUEST SENSE	No	03h	M	SPC-2[b]
RESERVE(6)	No	16h	M	SPC-2[b]
RESERVE(10)	No	56h	M	SPC-2[b]
REWIND	Yes	01h	M	5.3.11
SEND DIAGNOSTIC	Yes[a]	1Dh	M	SPC-2[b]
SPACE	May	11h	M	5.3.12
TEST UNIT READY	No	00h	M	SPC-2
VERIFY	Yes	13h	O	5.3.13
WRITE	No	0Ah	M	5.3.14
WRITE BUFFER	Yes[a]	3Bh	O	SPC-2[b]
WRITE FILEMARKS	May	10h	M	5.3.15

Key:
M = command implementation is mandatory
O = command implementation is optional
SPC-2 = SCSI Primary Commands – 2 standard
SMC = SCSI-3 Medium Changer Commands standard

[a] Refer to subclause 5.2.5.
[b] See ANSI INCITS 351-2001
[c] ISO/IEC 14776-351, see Bibliography.

5.3.1 ERASE command

The ERASE command (see Table 11) causes part or all of the medium to be erased beginning at the current position. Erased means the medium shall be erased or a pattern shall be written on the medium that appears as a gap. Prior to performing the erase operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 11 – 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							

An immediate (immed) bit of zero indicates that the device server shall not return status until the erase operation has completed. Interpretation of an immed bit of one depends on the value of the long bit, see below. However, for all values of the long bit, if CHECK CONDITION status is returned for an ERASE 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 data, filemarks and setmarks have been written to the medium 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 5 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 (see 5.4.3.2). 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 (see 5.3.14). If the long bit is zero, the erase operation shall terminate with CHECK CONDITION status and set the sense data as defined for the WRITE command. Any count of pending buffered erase shall not be reported as part of the value returned in the information field or in the READ POSITION response data.

5.3.2 FORMAT MEDIUM command

The FORMAT MEDIUM command (see Table 12) is used to prepare the medium for use by the logical unit. If buffered data, filemarks, or setmarks are stored by the device server when processing of a FORMAT MEDIUM command begins, the command shall be rejected with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier set to POSITION PAST BEGINNING OF MEDIUM.

Table 12 – FORMAT MEDIUM command

Bit / Byte	7	6	5	4	3	2	1	0
0	operation code (04h)							
1	Reserved						verify	immed
2	Reserved				format			
3	(MSB)	transfer length						
4								(LSB)
5	control							

The FORMAT MEDIUM command shall be accepted only when the medium is at beginning-of-medium (BOM) 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 set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier set to POSITION PAST BEGINNING OF MEDIUM.

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

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

In response to all commands except REQUEST SENSE and INQUIRY, the device server shall return CHECK CONDITION unless a reservation conflict exists. In that case RESERVATION CONFLICT status shall be returned; or

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 and an additional sense code qualifier set to LOGICAL UNIT NOT READY – FORMAT IN PROGRESS, with the sense key specific bytes set for process indication (as described in ANSI INCITS 351-2001).

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 format the medium and then 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 and an additional sense code qualifier set to MEDIUM FORMAT CORRUPTED. If the verify bit is zero, the logical unit shall not perform the verify check.

The format field is defined in Table 13.

Table 13 – Format field definition

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

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

If the format field is 1h, the logical unit shall partition the medium using the current mode data from medium partition mode pages(1-4) (see 5.4.3.3 and 5.4.3.4). If none of the mode bits sdp, fdp, or idp are set to one, the device server shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST with the addition sense code set to PARAMETER VALUE INVALID. If insufficient space exists on the medium for the requested partition sizes, the device server shall return CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR with the additional sense code and an additional sense code qualifier set to VOLUME OVERFLOW. A valid FORMAT MEDIUM command with 1h in the format field may cause all data on the entire physical volume to be lost.

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

When the format field contains 1h or 2h, some errors related to mode page field contents may not be detected until the FORMAT MEDIUM command is processed. Therefore, some error conditions described in 5.4.3.3 and 5.4.3.4 may be returned in response to a FORMAT MEDIUM command with 1h or 2h in the format field.

The transfer length specifies the length in bytes of format information that shall be transferred from the initiator. 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, 1h, or 2h, the transfer length shall be zero. Use of format information is restricted to vendor-specific values in the format field and the contents of the format information is vendor-specific.

5.3.3 LOAD UNLOAD command

The LOAD UNLOAD command (see Table 14) requests that the logical unit enable or disable the logical unit for further operations. This command may also be used to request a retension function. Prior to performing the LOAD UNLOAD operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium. If the buffered mode is not 0h (see 5.4.3) and a previous command was terminated with CHECK CONDITION status and the device is unable to continue writing successfully, the logical unit shall discard any unwritten buffered data, filemarks, and setmarks prior to performing the LOAD UNLOAD operation.

Table 14 – LOAD UNLOAD command

Bit Byte	7	6	5	4	3	2	1	0
0	operation code (1Bh)							
1	Reserved							immed
2	Reserved							
3	Reserved							
4	Reserved				hold	eot	reten	load
5	control							

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

NOTE 6 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.

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

A load bit of one and a hold bit of one indicates if the medium has not been moved into the logical unit, the medium shall be moved in but not positioned for access. The eot and reten bits shall be set to zero. Following successful completion, the device server shall return GOOD STATUS. If both the medium and device server support MAM, the device server shall generate a unit attention condition for all initiators with the additional sense code and additional sense code qualifier set to MEDIUM AUXILIARY MEMORY ACCESSIBLE.

A load bit of zero and a hold bit of one indicates if the medium is in the logical unit, the medium shall be positioned as specified by the reten and eot bits or shall be unthreaded (whichever is appropriate for the medium type) but shall not be ejected. Following successful completion, the device server shall return GOOD STATUS. If both the medium and device server support MAM, the device server shall generate a unit attention condition for all initiators with the additional sense code and additional sense code qualifier set to MEDIUM AUXILIARY MEMORY ACCESSIBLE.

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

An end-of-tape (eot) bit of one 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 with the additional sense code and additional sense code qualifier set to INVALID FIELD IN CDB.

A hold bit of one indicates MAM shall be accessible upon completion of the command but the medium shall not be positioned for access. A hold bit of zero and a load bit of one indicates the medium shall be positioned for access. A hold bit of zero and a load bit of zero indicates MAM shall not be accessible upon completion of the command.

If the autoload mode field in the control mode page (see ANSI INCITS 351-2001) is changeable or a MODE SENSE command reports a value in the autoload mode field other than zero, the device server shall support a value of one for the hold bit.

When operating in buffered mode 1h or 2h (see 5.4.3), the logical unit shall discard any unwritten buffered data after the LOAD UNLOAD command is validated if the device is unable to continue successfully writing (for example, the device reported CHECK CONDITION status to a previous command, reported a write-type error, and the error has not been cleared or otherwise recovered).

5.3.4 LOCATE command

LOCATE command (see Table 15) causes the logical unit to position the medium to the specified logical element with a matching block identifier in the specified partition. Upon completion, the logical position shall be before the specified logical element. 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 15 – 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 address							
5								
6								(LSB)
7	Reserved							
8	partition							
9	control							

A block identifier type (bt) bit of one indicates the value in the block address field shall be interpreted as a vendor-specific value. A bt bit of zero indicates the value in the block address field shall be interpreted as a block identifier (see 5.2.7).

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

An immediate (immed) bit of zero indicates that the device server shall not return status until the locate operation has completed. If the immed bit is one, the device server shall return status as soon as all buffered data, filemarks and setmarks have been written to the medium 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 specifies the block identifier 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 sense data 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 sense data eom bit shall be set to one.

The partition field specifies the partition to select if the cp bit is one. Refer to the sequential-access device model (see 5.2.3) and the medium partition pages (see 5.4.3.3 and 5.4.3.4.) for additional information about partitioning.

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

5.3.5 READ command

The READ command (see Table 16) requests that the device server transfer one or more block(s) of data to the application client beginning with the next block. 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 16 – 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							

The fixed bit specifies whether fixed-length or variable-length blocks are to be transferred. Refer to the READ BLOCK LIMITS command (see 5.3.6) 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 ANSI INCITS 351-2001). 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 ANSI INCITS 351-2001); or

b) not report CHECK CONDITION status if the only error is the underlength condition, or if the only error is the overlength condition and the block length field of the mode parameters block descriptor is zero.

NOTE 7 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 (for example, 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 and an additional sense code qualifier 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 with an additional sense code and an additional sense code qualifier of NO ADDITIONAL SENSE INFORMATION. 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 8 In the above case with the fixed bit of one, only the position of the incorrect-length logical block may be determined from the sense data. The actual length of the incorrect block is not reported. Other means may be used to determine its actual length (for example, 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 with an additional sense code and an additional sense code qualifier of FILEMARK DETECTED. Upon termination, the logical position shall be after the filemark (end-of-partition side). If the fixed bit is one, the information field shall be set to the requested transfer length minus the actual number of 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 5.4.3.2), 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 and an additional sense code qualifier 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 5.4.3.2), 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 with an additional sense code and an additional sense code qualifier of END-OF-PARTITON/MEDIUM DETECTED. The eom and valid bits shall be set to one in the sense data. Upon termination, the logical position shall be after the last block transferred (end-of-partition side). If the fixed bit is one, the information field in the sense data shall be set to the requested transfer length minus the actual number of blocks read. If the fixed bit is zero, the information field in the sense data shall be set to the requested transfer length minus the actual 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 9 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 immediately after the last recorded logical record, filemark, or setmark (end-of-partition side). If the fixed bit is one, the information field in the sense data shall be set to the requested transfer length minus the actual number of 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 in the sense data shall be set to the requested transfer length.

5.3.6 READ BLOCK LIMITS command

The READ BLOCK LIMITS command (see Table 17) requests that the block length limits capability of the logical unit be returned. The READ BLOCK LIMITS data (see Table 18) shall be returned.

Table 17 – READ BLOCK LIMITS command

Bit / Byte	7	6	5	4	3	2	1	0
0	operation code (05h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	control							

Table 18 – READ BLOCK LIMITS data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved			granularity				
1	(MSB)	maximum block length limit						
2								
3								(LSB)
4	(MSB)	minimum block length limit						
5								(LSB)

The granularity field indicates the supported block size granularity. The logical unit shall support all block sizes n such that n minus the minimum block length limit is a multiple of 2GRANULARITY and n is greater than or equal to the minimum block length limit and less than or equal to the maximum block length limit.

If the maximum block length limit value equals the minimum block length limit value, the logical unit supports the transfer of data in the fixed-block mode only, with the block length equal to the minimum block length limit value. In this case, READ and WRITE commands with the fixed bit set to zero shall result in CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier set to INVALID FIELD IN CDB.

In fixed-block mode the value of maximum block length and minimum block length shall be a multiple of four.

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.

5.3.7 READ POSITION command

The READ POSITION command (see Table 19) reports the current position and provides information about any data blocks, filemarks and/or setmarks in the buffer. No medium movement shall occur as a result of responding to the command.

Table 19 – 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 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 a tclp value of one 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 and an additional sense code qualifier 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 bit of zero indicates the device server shall return 20 bytes of data.

The long bit and the tclp bit shall be equal. If the long and tclp bits are not equal, or if both the long and the block identifier 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 and an additional sense code qualifier set to INVALID FIELD IN CDB.

The block identifier type (bt) controls the content of the short format data. A bt bit of one requests the device server to return the first block location, the last block location, and block number fields as vendor-specific values. A bt bit of zero requests the device server to return the first block location, the last block location, and block number fields as block identifier values (see 5.2.7), (relative to a partition).

Table 20 indicates the READ POSITION data that shall be returned if the tclp and long bits are set to zero.

Table 20 – 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	(MSB)							
:				first block location				
7								(LSB)
8	(MSB)							
:				last block location				
11								(LSB)
12	Reserved							
13	(MSB)							
:				number of blocks in buffer				
15								(LSB)
16	(MSB)							
:				number of bytes in buffer				
19								(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.

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 currently known or not otherwise obtainable. A bpu bit of zero indicates that the first block location and last block location fields contain valid position information.

A position error (perr) bit of one indicates that the logical unit is unable to report the correct position due to an overflow of any of the returned position data. A perr bit of zero 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 identifier associated with the current logical position. The value shall indicate the block identifier 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 identifier (see 5.2.7) associated with the next block to be transferred from the buffer to the medium. The value shall indicate the block identifier 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 10 The information provided by the first block location and last block location fields may 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 buffer of the logical unit that have not been written to the medium.

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

Table 21 indicates the format of the READ POSITION data that shall be returned if the tclp and long bits are set to one.

Table 21 – 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	(MSB)							
:				partition number				
7								(LSB)
8	(MSB)							
:				block number				
15								(LSB)
16	(MSB)							
:				file number				
23								(LSB)
24	(MSB)							
:				set number				
31								(LSB)

The bop, eop, and partition number fields are as defined in the READ POSITION data returned when the tclp bit is set to zero.

A block position unknown (bpu) bit of one indicates that the partition number or block number are not known or accurate reporting is not currently available. A bpu bit of zero indicates that the partition number and block number fields contain valid position information.

A mark position unknown (mpu) bit of one indicates the file number and set number are not known or accurate reporting is not currently available. An mpu bit of zero indicates the file number and set number fields contain valid position information.

The mpu and bpu bits shall be set to one only if the logical unit is unable to accurately assume or does not know the current mark or block, respectively.

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 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.

NOTE 11 The reported set number value is not affected by the value of the rsmk bit in the device configuration page.

5.3.8 READ REVERSE command

The READ REVERSE command (see Table 22) requests that the device server transfer one or more block(s) of data to the application client beginning at the current logical position. Prior to performing the read reverse operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 22 – READ REVERSE command

Bit / Byte	7	6	5	4	3	2	1	0
0	operation code (0Fh)							
1	Reserved					bytord	sili	fixed
2	(MSB)							
3	transfer length							
4								(LSB)
5	control							

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

A byte order (bytord) bit of zero indicates that 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. A bytord bit of one indicates that all block(s) are transferred in the reverse order but the byte(s) within the block(s) are transferred in the same order as returned by the READ command. Support for either value of the bytord bit is optional.

Refer to the READ command (see 5.3.5) for a description of the fixed bit, the sili bit, the transfer length field, and any conditions that may result from incorrect usage of these fields.

Filemarks, 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.

5.3.9 RECOVER BUFFERED DATA command

The RECOVER BUFFERED DATA command (see Table 23) 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 the buffered data after error or exception conditions make it impossible to write the buffered data to the medium. One or more RECOVER BUFFERED DATA commands may be required to recover all unwritten buffered data.

Table 23 – RECOVER BUFFERED DATA command

Bit Byte	7	6	5	4	3	2	1	0
0	operation code (14h)							
1	Reserved						sili	fixed
2	(MSB)							
3				transfer length				
4								(LSB)
5	control							

The 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 that block(s) are transferred is defined by the rbo bit in the device configuration page (see 5.4.3.2). 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 5.3.5) for a description of the fixed bit, the sili bit, the transfer length field, and any conditions that may result from incorrect usage of these fields.

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

NOTE 12 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 5.4.3.2), 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 and an additional sense code qualifier 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, the additional sense code and an additional sense code qualifier shall bet set to END-OF-DATA DETECTED, and the eom and valid bits shall be set to one in the sense data. If the fixed bit is one, the information field shall be set to the requested transfer length minus the actual number of blocks transferred. If the fixed bit is zero, the information field shall be set to the requested transfer length.

5.3.10 REPORT DENSITY SUPPORT command

The REPORT DENSITY SUPPORT command (see Table 24) requests that information regarding the supported densities for the logical unit be sent to the application client.

Table 24 – REPORT DENSITY SUPPORT command

Bit / Byte	7	6	5	4	3	2	1	0
0	operation code (44h)							
1	Reserved							medium
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)			allocation length				
8								(LSB)
9	control							

A medium bit of zero indicates that the device server shall return density support data block descriptors for densities supported by the logical unit for any supported medium. A medium bit of one indicates that the device server shall return density support data block descriptors for densities supported by the mounted medium. If the medium 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 and the additional sense code and an additional sense code qualifier shall indicate the reason for 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 25) followed by one or more density support data block descriptors (see Table 26). The density support data block descriptors shall follow the density support header. The density support data block descriptors shall be in numerical ascending order of the primary density code value.

Table 25 – Density support header

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	available density support length						
1								(LSB)
2	Reserved							
3	Reserved							
4 *n*	density support data block descriptors							

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 (the length in bytes of a density support data block descriptor).

Table 26 – Density support data block descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	primary density code							
1	secondary density code							
2	wrtok	dup	deflt	Reserved				
3	Reserved							
4	Reserved							
5	(MSB)	bits per mm						
:								
7								(LSB)
8	(MSB)	medium width						
9								(LSB)
10	(MSB)	tracks						
11								(LSB)
12	(MSB)	capacity						
:								
15								(LSB)
16 : 23	assigning organization							
24 : 31	density name							
32 : 51	description							

Density support data block descriptors shall be returned by ascending primary density code values. Multiple entries may exist for a given primary density code value. For all entries with equal primary density code values, all fields except for assigning organization, density name, and description shall be identical. Density support data block descriptors with the same primary density code value should be ordered from most to least preferred assigning organization, density name, and description.

NOTE 13 By allowing multiple entries for a given primary and secondary density code set, multiple standard names may identify the same density code. This facilitates the remapping of density codes, if required.

The density support data block descriptor may represent a particular format in addition to giving physical density information. The information in a density support data block descriptor provides an application client with a detailed review of the recording technologies supported by a logical unit. By supplying the density code value returned in a density support data block descriptor in a MODE SELECT command (see 5.4.3), an application client selects the recording technology (density, format, etc.).

The primary density code field contains the value returned by a MODE SENSE command for the density described in the remainder of the density support data block descriptor. The device server shall accept a MODE SELECT command containing this value for an appropriate medium. 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. When density information matches one of the entries in Table A.1, the primary density code value should match the density code assigned in the table.

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

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

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

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. If either the primary density code or the secondary density code field is zero, the deflt bit shall be one. If neither the primary or secondary density code is zero and the deflt bit is one, the logical unit shall accept a MODE SELECT header with a density code of 00h as equivalent to the primary and secondary density codes.

NOTE 14 The default density of the logical unit may vary depending on the currently mounted medium. Multiple codes may return a deflt bit of one when the medium 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 medium 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 medium 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 value may vary for a given density depending on the mounted medium. Direct comparison of this value between different vendors (possibly products) is discouraged since the definition of the number of tracks may vary. For recording formats that are neither parallel

nor serpentine, the tracks field indicates the maximum number of data tracks that are read or recorded simultaneously.

If the medium bit is zero, the capacity field shall indicate the approximate capacity of the longest supported medium assuming recording in this density with one partition. If the medium bit is one, 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 is not available 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 medium is in "good" condition, and that "normal" data and block sizes are used. This value is in units of megabytes (106 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 medium and the method used to measure maximum capacity may vary. The capacity field is intended to be used by the application client to determine that the correct density is being used, particularly when a lower-density format is required for interchange.

The assigning organization field contains eight bytes of ASCII data identifying the organization responsible for the specifications defining the values in this density support data block descriptor. The data shall be left aligned within this field. The ASCII value for a space (20h) shall be used if padding is required. The assigning organization field should contain a value listed in the vendor identification list (see ANSI INCITS 351-2001). The use of a specific vendor identification, other than the one associated with the device is allowed.

NOTE 15 If vendor X defines a density and format, another vendor may use X in the assigning organization field. If exactly the same density and format construction later becomes known by another name, both X and the new assigning organization may be used for the density code. This is one condition that may result in multiple density support data block descriptors for a single density code value.

NOTE 16 It is intended that the assigning organization field contain a unique identification of the organization responsible for the information in a density support data block descriptor. In the absence of any formal registration procedure, T10 maintains a list of vendor and assigning organization identification codes in use. Vendors are requested to voluntarily submit their identification codes to prevent duplication of codes.

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

NOTE 17 It is suggested that any document which specifies a format and density for the medium contain the values to be used by a logical unit when reporting the density support. The values for the bits per mm, medium 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.

5.3.11 REWIND command

The REWIND command (see Table 27) 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. If the buffered mode is not 0h (see 5.4.3) and a previous command was terminated with CHECK CONDITION status and the device is unable to continue successfully writing, the logical unit shall discard any unwritten buffered data, filemarks and setmarks prior to performing the REWIND operation.

Table 27 – REWIND command

Bit / Byte	7	6	5	4	3	2	1	0
0	operation code (01h)							
1	Reserved							immed
2	Reserved							
3	Reserved							
4	Reserved							
5	control							

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

NOTE 18 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.

5.3.12 SPACE command

The SPACE command (see Table 28) provides a variety of positioning functions that are determined by the code and count fields. Both forward and reverse positioning are provided, although some logical units may only support a subset of this command. 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 REQUEST and the additional sense code and an additional sense code qualifier shall be set to INVALID FIELD IN CDB. 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. The information field value shall be equal to the magnitude of the count field minus the magnitude of the blocks, filemarks or setmarks spaced over. A CHECK CONDITION caused by early termination of any SPACE command shall not result in a negative information field value.

Table 28 – 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							

The code field is defined in Table 29.

Table 29 – Code 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 to 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 field and the code field is not 011b (End-of-data) 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, if they exist. A zero value in the count field and the code field is not 011b (End-of-data) shall cause no change of logical position. A negative value *–N* (two's complement notation) in the count field and the code field is not 011b (End-of-data) 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, if they exist. When the code field is 011b (End-of-data), the COUNT field shall be ignored and the device server shall transfer any buffered data, filemarks and setmarks to the medium before moving before the End-of data position. When the count field is zero and the code field is not 011b (End-of-data) a device server is not required to transfer any buffered data, filemarks, and setmarks to the medium. Support of spacing in the reverse direction is optional.

If a filemark is encountered while spacing over blocks, the command shall be terminated. CHECK CONDITION status shall be returned, and the filemark and valid bits shall be set to one in the sense data. The sense key shall be set to NO SENSE and the additional sense code and an additional sense code qualifier shall be set to FILEMARK DETECTED. The information field shall be set to the requested count minus the actual number of blocks spaced over (not including the filemark). The logical position shall be on the end-of-partition side of the filemark if movement was in the forward direction and on the beginning-of-partition side of the filemark if movement was in the reverse direction.

If a setmark is encountered while spacing over blocks or filemarks and the rsmk bit is set to one in the device configuration page (see 5.4.3.2), the command shall be terminated, CHECK CONDITION status shall be returned, and the filemark and valid bits shall be set to one in the sense data. The sense key shall be set to NO SENSE and the additional sense code and an additional sense code qualifier 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 5.4.3.2), 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 additional sense code and an additional sense code qualifier shall be set to END-OF-PARTITION/MEDIUM DETECTED. 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 19 Setting the rew bit to one 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 sense data valid bit shall be set to one in the sense data. The additional sense code and an additional sense code qualifier shall be set to END-OF-DATA DETECTED. The sense data eom bit shall be set to one if end-of-data is encountered at or after early-warning. The information field shall be set to the requested count minus the actual number of blocks, filemarks or setmarks spaced over as defined by the code value. The medium shall be positioned such that a subsequent write operation would append to the last record, filemark or setmark.

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 additional sense code and an additional sense code qualifier shall be set to END-OF-PARTITION/MEDIUM DETECTED, and the sense data eom and 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 additional sense code and an additional sense code qualifier shall be set to BEGINNING-OF-PARTITION/MEDIUM DETECTED. The sense data eom and valid bits shall be set to one, and the information field set to the total number of 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; or
c) a negative value –*N* (two'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 5.4.3.2), CHECK CONDITION status shall be returned, the filemark bit shall be set to one and the valid bit shall be set to zero in the sense data. The sense key shall be set to NO SENSE and the additional sense code and an additional sense code qualifier 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 setmarks 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 additional sense code and an additional sense code qualifier shall be set to END-OF-PARTITION/MEDIUM DETECTED, the eom bit shall be set to one and the valid bit shall be set to zero in the sense data.

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, the additional sense code and an additional sense code qualifier shall be set to END-OF-DATA DETECTED, and the sense data valid bit shall be set to zero. The medium shall be positioned such that a subsequent write operation would append to the last record, filemark or setmark. The sense data eom bit shall be set to one if end-of-data is encountered at or after early-warning.

When spacing to end-of-data, the count field is ignored. Upon successful completion, the medium shall be positioned such that a subsequent write operation would append to the last record, filemark or setmark.

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 additional sense code and an additional sense code qualifier shall be set to END-OF-PARTITION/MEDIUM DETECTED, the eom bit shall be set to one, and the valid bit shall be set to zero in the sense data.

5.3.13 VERIFY command

The VERIFY command (see Table 30) requests that the device server verify one or more block(s) beginning at the current logical position. Prior to performing the verify operation, the logical unit shall ensure that all buffered data, filemarks, and setmarks have been transferred to the medium.

Table 30 – 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							

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 verification data has been transferred from the initiator to the device server, if the bytcmp bit is one).

NOTE 20 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 (for example, 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 (see 5.3.14). If the bytcmp bit is one and the byte compare option is not supported, the device server shall terminate the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code and an additional sense code qualifier shall be set to INVALID FIELD IN CDB.

The verification length field specifies the amount of data to verify, in blocks or bytes, as indicated by the fixed bit. Refer to the READ command (see 5.3.5) for a description of the fixed bit and any error conditions that may result from incorrect usage. If the bytcmp bit is one and the verification length field is zero, no data shall be verified and the current logical position shall not be changed. This condition shall not be considered as an error.

The VERIFY 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 5.4.3.2),

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 5.4.3.2), 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 5.3.5). 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 sense data valid bit shall be set to one the sense key shall be set to MISCOMPARE, and the additional sense code and an additional sense code qualifier set to MISCOMPARE DURING VERIFY OPERATION. If the fixed bit is one, the information field shall be set to the requested verification length minus the actual number of 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).

5.3.14 WRITE command

The WRITE command (see Table 31) requests that the device server write the data that is transferred from the application client to the current logical position.

Table 31 – 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							

The fixed bit specifies whether fixed-length or variable-length blocks are to be transferred. See the READ BLOCK LIMITS command (see 5.3.6) for additional information about fixed and variable block mode.

If the fixed bit is one, the transfer length value specifies the number of fixed-length blocks to be transferred, using the current block length reported in the mode parameter block descriptor (see ANSI INCITS 351-2001). If the fixed bit is zero, a single block is transferred with transfer length specifying the block length in bytes.

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

A WRITE command may be buffered or unbuffered, as indicated by the buffered mode field of the mode parameter header (see 5.4.3). When operating in unbuffered mode (see 5.1.21), the device server shall not return GOOD status until all data block(s) are successfully written to the medium. When operating in buffered mode (see 5.1.3), the device server may return GOOD status as soon as all data block(s) are successfully transferred to the logical unit's buffer.

NOTE 21 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 5.4.3.2). The command shall terminate with CHECK CONDITION status, the additional sense code and an additional sense code qualifier shall be set to END-OF-PARTITION/MEDIUM DETECTED, 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 the device server is unable to transfer any data, buffered or unbuffered, when early-warning is encountered, the sense key shall be set to VOLUME OVERFLOW.

The information field shall be defined as follows:

a) if the device is operating in unbuffered mode (see 5.1.21) 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 the device is operating in unbuffered mode and the fixed bit is set to zero, the information field shall be set to the requested transfer length;

c) if the device is operating in buffered mode (see 5.1.3) 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. The value in the information field may exceed the transfer length; and

d) if the device is operating in buffered mode 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 22 The logical unit should ensure that some additional data may be written to the medium (for example, labels, filemarks or setmarks) after the first early-warning indication has been returned to the application client (see 5.2.2).

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 in the sense data 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 in the sense data shall be defined as follows:

a) if the fixed bit is one, the information field in the sense data shall be set to the requested transfer length minus the actual number of blocks written to the medium; or

b) if the fixed bit is zero, the information field in the sense data shall be set to the requested transfer length.

NOTE 23 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 may cause confusion as to whether there was really a 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 initiator) shall be discarded. 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 medium.

NOTE 24 Repositioning of the medium may be required to remove a logical block from the medium if part of the block is written to the medium before being completely transferred to the device server. While vendor-specific, a period of time may exist in which the partial block is not overwritten. Attempting to write beyond 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). The application client should issue a READ POSITION command to determine the logical position after a write operation is aborted.

5.3.15 WRITE FILEMARKS command

The WRITE FILEMARKS command (see Table 32) requests that the device server write the specified number of filemarks or setmarks to the current position.

Table 32 – 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							

If the write setmark (wsmk) bit is one, the transfer length field specifies the number of setmarks to be written. If the wsmk bit is zero, the transfer length field specifies the number of filemarks to be written. It shall not be considered an error for the transfer length field to contain zero. Support of wsmk set to one is optional.

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 the device is operating in buffered mode (see 5.1.3).

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 25 Upon completion of any buffered write operation, the application client may 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 5.4.3.2). The command shall terminate with CHECK CONDITION status, the additional sense code and an additional sense code qualifier shall be set to END-OF-PARTITION/MEDIUM DETECTED, 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 the device is operating in unbuffered mode (see 5.1.21), the information field shall be set to the requested transfer length minus the actual number of filemarks or setmarks written;

b) if the device is operating in buffered mode (see 5.3.14) and the buffered data was written in variable block mode (see 5.3.14), 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). It is possible for the value in the information field to exceed the transfer length; or

c) if the device is operating in buffered mode and the buffered data was written in fixed block mode (see 5.3.14), the information field shall be set to the total number of blocks, filemarks and setmarks not written (the number of filemarks or setmarks not transferred from the application client plus the number of blocks, filemarks and setmarks remaining in the logical unit's buffer). It is possible for the value in the information field to exceed the transfer length.

NOTE 26 The logical unit should ensure that some additional data may be written to the medium (for example, labels, filemarks, or setmarks) after the first early-warning indication has been returned to the application client (see 5.2.2).

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.

5.4 Parameters for sequential-access devices

5.4.1 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 33.

Table 33 – Diagnostic page codes

Page code	Description	Reference
00h	Supported diagnostic pages	SPC-2[a]
01h to 3Fh	Reserved (for all device type pages)	
40h to 7Fh	Reserved	
80h to FFh	Vendor-specific pages	
[a] See ANSI INCITS 351-2001.		

5.4.2 Log parameters

5.4.2.0 General

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 34.

Table 34 – Log page codes

Page code	Description	reference
01h	Buffer over-run/under-run page	SPC-2[a]
04h	Error counter page (read reverse)	SPC-2[a]
03h	Error counter page (read)	SPC-2[a]
05h	Error counter page (verify)	SPC-2[a]
02h	Error counter page (write)	SPC-2[a]
08h	Format status page	SPC-2[a]
0Bh	Last *n* deferred error events page	SPC-2[a]
07h	Last *n* error events page	SPC-2[a]
06h	Non-medium error page	SPC-2[a]
09h to 0Ah	Reserved	
0Fh to 2Dh	Reserved	
2Fh	Reserved	
0Ch	Sequential-access device page	5.4.2.1
0Eh	Start-stop cycle counter page	SPC-2[a]
00h	Supported log pages	SPC-2[a]
2Eh	TapeAlert log page	5.4.2.2
0Dh	Temperature page	SPC-2[a]
30h to 3Fh	Vendor-specific (does not require page format)	

[a] See ANSI INCITS 351-2001.

5.4.2.1 Sequential-access device page

The sequential-access device page (page code 0Ch) defines data counters associated with data bytes transferred to and from the medium 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 27 The data in parameters 0 and 1 are intended to provide an indication of the compression ratio for the written data. Parameters 2 and 3 are intended to provide an indication of the compression ratio for read data.

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

Table 35 defines the parameter codes for the sequential-access device page.

Table 35 – 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 medium as a result of WRITE command operations, not counting ECC and formatting overhead.
0002h	Number of data bytes read from the medium during READ command operations, not counting ECC and formatting overhead.
0003h	Number of data bytes transferred to the initiator(s) during READ command operations.
0004h – 00FFh	Reserved
0100h	Cleaning required
0101h to 7FFFh	Reserved
8000h to 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.

5.4.2.2 TapeAlert log page

The TapeAlert log page (2Eh) defines error and informational flags used for detailed device diagnostics (see Table 36). Refer to Table B.1 for a description of the flags. Refer to ANSI INCITS 351-2001 for a definition of the log page fields.

Table 36 – TapeAlert log page

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved			Page code (2Eh)				
1	Reserved							
2	(MSB)	Parameter length (140h)						
3								(LSB)
$5n - 1$	(MSB)	Parameter code (n)						
$5n$								(LSB)
$5n + 1$	du (0)	ds (1)	tsd (0)	etc (0)	tmc (0)		lbin (0)	lp (0)
$5n + 2$	Parameter length (1)							
$5n + 3$	Value of flag							

NOTE 1 The value of n in the Parameter code field shall be in the range of 1 to 64.

NOTE 2 A value of one in the Value of flag field indicates the flag is set.

5.4.3 Mode parameters

5.4.3.0 General

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 ANSI INCITS 351-2001.

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

The device-specific parameter field in the mode parameter header is defined in Table 37 for sequential-access devices.

Table 37 – Device-specific parameter

Bit Byte	7	6	5	4	3	2	1	0
	wp	buffered mode			speed			

When used with the MODE SENSE command, a write protect (wp) bit of zero indicates that a 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 ignored.

NOTE 28 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 38.

Table 38 – 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: all the data specified in the write command has been successfully transferred to the logical unit's buffer, and all buffered data from different initiators has been successfully written to the medium.
3h to 7h	Reserved

Values for the speed field shall be assigned as defined in Table 39.

Table 39 – Speed field definition

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

For the MODE SELECT command, the density code field of the sequential-access device block descriptor (see ANSI INCITS 351-2001) 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 mounted or because the density of the installed medium has not been determined, the density code field should be set to the principal density code value (see 5.1.15). 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 is unable to automatically determine the density from the medium, or

   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

   4) 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, or

   5) 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 40 lists the sequential-access device density codes.

Table 40 – Sequential-access density codes

Code value	Description	Note
00h	Default density	1
01h to 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	

NOTE 1 Only reported by MODE SENSE commands if primary density code for the density is given.

NOTE 2 See informative Annex A for density codes.

NOTE 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 41.

Table 41 – Mode page codes

Page code	Description	Reference
0Ah	Control mode page	SPC-2[a]
0Fh	Data compression page	5.4.3.1
10h	Device configuration mode page	5.4.3.2
02h	Disconnect-reconnect page	SPC-2[a]
1Ch	Informational exceptions control mode page	5.4.3.6
11h	Medium partition mode page(1)	5.4.3.3
12h	Medium partition mode page(2)	5.4.3.4
13h	Medium partition mode page(3)	5.4.3.4
14h	Medium partition mode page(4)	5.4.3.4
09h	Obsolete	3.3.7
1Ah	Power condition mode page	SPC-2[a]
18h	Protocol specific LUN mode page	SPC-2[a]
19h	Protocol specific port mode page	SPC-2[a]
01h	Read-write error recovery page	5.4.3.5
03h to 08h	Reserved	
0Bh to 0Eh	Reserved	
15h to 17h	Reserved	
1Bh	Reserved	
1Dh to 1Fh	Reserved	
3Fh	Return all pages (valid only for the MODE SENSE command)	
00h	Vendor-specific (does not require page format)	
20h to 3Eh	Vendor-specific (does not require page format)	
[a] See ANSI INCITS 351-2001.		

5.4.3.1 Data compression page

The data compression page (see Table 42) specifies the parameters for the control of data compression in a sequential-access device.

Table 42 – 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	(MSB)	compression algorithm						
:								
7								(LSB)
8	(MSB)	decompression algorithm						
:								
11								(LSB)
12	Reserved							
13	Reserved							
14	Reserved							
15	Reserved							

The parameters savable (ps) bit is only used for the MODE SENSE command. This bit is reserved for 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.

A data compression enable (dce) bit of one indicates that data compression is enabled. When this bit is one, data sent to the device server by the application client shall be processed using the selected compression algorithm before being written to the medium. A dce bit of zero indicates that data compression is 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. A dcc bit of zero indicates that the device does not support data compression. This shall be a non-changeable bit.

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

The report exception on decompression (red) field indicates the response to certain boundaries it detects in the data on the medium. There are a number of boundaries that may occur on the medium between compressed and uncompressed data. These boundaries are shown in Table 43. Only boundaries shown in Table 43 may generate a CHECK CONDITION status.

Table 43 – Possible boundaries and resulting sense keys due to data compression

Prior data	Current data	Sense key (see notes 1, 2)		
		RED = 0	RED = 1	RED = 2
Uncompressed	Compressed (unsupported algorithm)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
Uncompressed	Compressed (supported algorithm)	[none]	[none]	RECOVERED ERROR
Compressed (supported algorithm)	Uncompressed	[none]	[none]	NO SENSE
Compressed (supported algorithm)	Compressed (unsupported algorithm)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
Compressed (supported algorithm A)	Compressed (supported algorithm B)	[none]	[none]	RECOVERED ERROR
Compressed (unsupported algorithm)	Uncompressed	[none]	NO SENSE	NO SENSE
Compressed (unsupported algorithm)	Compressed (supported algorithm)	[none]	RECOVERED ERROR	RECOVERED ERROR
Compressed (unsupported algorithm A)	Compressed (unsupported algorithm B)	MEDIUM ERROR	MEDIUM ERROR	MEDIUM ERROR
All other combinations		[none]	[none]	[none]

NOTE 1 [none] indicates that no CHECK CONDITION status is returned given the data boundary condition and the current value of red.

NOTE 2 The appropriate additional sense code and an additional sense code qualifier is specified below.

If a CHECK CONDITION status is returned and the current data is compressed, the additional sense code shall be set to either DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN with the additional sense code qualifier set to the algorithm id or DECOMPRESSION EXCEPTION LONG ALGORITHM with no additional sense code qualifier.

If a CHECK CONDITION status is returned and the current data is uncompressed, the additional sense code shall be set to DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN with the additional sense code qualifier set to zero.

A red field of zero indicates that the device shall return a CHECK CONDITION status when data is encountered on the medium during a read operation that the device is unable to decompress. Data boundaries in Table 43 marked other than [none] in the red = 0 column 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 that requires different handling by the application client than the data most recently encountered during a prior read operation. At each of these boundaries, the data that is sent to the application client is of a fundamentally different nature from that which was previously sent. Data boundaries in Table 43 marked other than [none] in the red = 1 column 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 that has been processed using a different algorithm from that data most recently encountered during a prior read operation. Data boundaries in Table 43 marked other than [none] in the red = 2 column shall generate CHECK CONDITION status with the specified sense key when the red field is two.

A red field of three is reserved. If a mode page containing a red field of three is received, the MODE SELECT command shall be terminated with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code and an additional sense code qualifier shall be set to INVALID FIELD IN PARAMETER LIST.

Upon detection of any of the boundary conditions described in Table 43 that results in a CHECK CONDITION status, the additional sense code and an additional sense code qualifier shall be set to either DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN (if the algorithm identifier is less than or equal to 255) or DECOMPRESSION EXCEPTION LONG ALGORITHM ID. The device shall, in both cases, set the decompression algorithm field to the algorithm identifier of the compression algorithm used to process the encountered data. The logical position shall be on the EOP side of the encountered data, and the information field in the sense data shall contain a count of the number of data blocks contained within the encountered data.

NOTE 29 When compressed data is encountered on the medium that the device is unable to 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 currently selected compression algorithm. The default value of the compression algorithm field shall indicate the default compression algorithm for the device. The 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 that the device does not support, then the device shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code and an additional sense code qualifier shall be set to INVALID FIELD IN PARAMETER LIST. Algorithm identifiers are shown in Table 43. The select data compression algorithm field in the device configuration page shall be ignored if a data compression page with the DCE bit set to one is also received by the device.

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 that was used to process the data encountered on the medium.

For the MODE SENSE command, the decompression algorithm field reflects the algorithm selected by the application client. For some devices, the decompression algorithm value returned in response to a MODE SENSE command may change dynamically to match the compression algorithm, detected by the device, that was used to process the data most recently encountered on the medium, during a read operation. A value of zero shall indicate that the data encountered on the medium during the most recent read operation was uncompressed. Compression algorithm identifiers are shown in Table 44.

Table 44 – Compression algorithm identifiers

Algorithm identifier	Description
00h	No algorithm selected (identifies uncompressed data).
01h	Set with MODE SELECT to select the default algorithm. MODE SENSE shall return the actual compression algorithm that was selected.
02h	Reserved
03h	IBM ALDC data compression algorithm with 512 byte buffer
04h	IBM ALDC data compression algorithm with 1 024 byte buffer
05h	IBM ALDC data compression algorithm with 2 048 byte buffer
06h to 0Fh	Reserved
10h	IBM IDRC data compaction algorithm
11h to 1Fh	Reserved
20h	DCLZ data compression algorithm
21h to FEh	Reserved
FFh	Unregistered algorithm
100h to FFFFFFFFh	Reserved

5.4.3.2 Device configuration page

The device configuration page (see Table 45) is used to specify the appropriate sequential-access device configuration.

Table 45 – 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	(MSB)	buffer size at early warning						
:								
13								(LSB)
14	select data compression algorithm							
15	Reserved					asocwp	perswp	prmwp

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.

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 the recording format that is in use for the selected density code when reading or writing data on a logical unit. The value of the active format field is vendor-specific.

The active partition field indicates the current logical partition number in use on the medium.

The write buffer full ratio field, on WRITE commands, indicates to the device server 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 field, on READ commands, indicates to the device server 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 field indicates the maximum time, in 100 ms increments, that the device server should wait before any buffered data that is to be written, is forced to the medium after the last buffered WRITE command that did not cause the buffer to exceed the buffer full ratio. A value of zero indicates that the device server 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. Most device servers consider this bit to be not changeable.

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. Most device servers consider this bit to be not changeable.

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 device server shall pre-read data from the medium in buffered mode to the limits of the buffer capacity without regard for filemarks. Values 01b, 10b and 11b specify that the device server shall terminate the pre-read operation if one, two, or three consecutive filemarks are detected, respectively. If the rsmk bit is one, the device server 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 buffer of the logical unit on a RECOVER BUFFERED DATA command in LIFO order (last-in-first-out) with respect to the order in 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 additional sense code and an additional sense code qualifier set to END-OF-PARTITION/MEDIUM DETECTED, and the eom bit set to one in the sense data when the early-warning position is encountered during read and write operations. If the rew bit is one and the sew bit is zero, the device server shall return CHECK CONDITION status with the sense key set to VOLUME OVERFLOW when early-warning is encountered during write operations.

NOTE 30 A rew bit of one is intended for compatibility with the 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 devices.

The gap size 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 eod defined field indicates the format type that the logical unit shall use to detect and generate the eod area. The values for eod defined are specified in Table 46.

Table 46 – EOD defined values

Code	Description
000b	Logical unit's default EOD definition
001b	Format-defined erased area of medium
010b	As specified in the SOCF field
011b	EOD recognition and generation is not supported
100b to 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 field. A value of zero indicates that EOD generation is disabled.

NOTE 31 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 5.3.14 and 5.3.15).

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 (see 5.2.9 and 5.2.9.2). When swp is one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status with the additional sense code and an additional sense code qualifier 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, that the logical unit shall reduce its logical buffer size to 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 vendor-specific.

NOTE 32 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. The select data compression algorithm field shall be ignored if a data compression page with the DCE bit set to one is also received by the device.

The associated write protect (asocwp) bit of one indicates the currently mounted volume is logically write protected until the volume is demounted (see 5.2.9 and 5.2.9.3). When asocwp is one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status and the sense key shall be set to WRITE PROTECT with the additional sense code and an additional sense code qualifier set to WRITE PROTECTED. An asocwp bit of zero indicates that the currently mounted volume is not write protected by the associated write protection. The asocwp bit shall be set to zero by the device server when the volume is demounted. This change of state shall not cause a unit attention condition. If the application client sets the asocwp bit to one while no volume is mounted, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense code and an additional sense code qualifier shall be set to LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED. If the device configuration page is savable, the asocwp bit shall be saved as zero, regardless of the current setting.

The persistent write protect (perswp) bit of one indicates the currently mounted volume is logically write protected (see 5.2.9 and 5.2.9.4). When perswp is one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status. The sense key shall be set to WRITE PROTECT. The additional sense code and an additional sense code qualifier shall be set to DATA PROTECT. An perswp bit of zero indicates that the currently mounted volume is not write protected by the persistent write protection. The perswp bit shall be set to zero by the device server when the volume is demounted or when a volume is mounted with persistent write protection disabled. The perswp shall be set to one by the device server when a volume is mounted with persistent write protection enabled. These changes of state shall not cause a unit attention condition. If the application client sets the perswp bit to one while no volume is mounted, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to NOT READY. The additional sense information shall be set to LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED. If the application client sets the perswp bit to one when the logical position is not at BOP 0, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST. The additional sense information shall be set to POSITION PAST BEGINNING OF MEDIUM. If the device configuration page is savable, the perswp bit shall be saved as zero, regardless of the current setting.

The permanent write protect (prmwp) bit of one indicates the currently mounted volume is logically write protected (see 5.2.9 and 5.2.9.5). When prmwp is one, all commands requiring eventual writes to the medium shall return CHECK CONDITION status and the sense key shall be set to WRITE PROTECT with the additional sense code and an additional sense code qualifier set to WRITE PROTECTED. An prmwp bit of zero indicates that the currently mounted volume is not write protected by the permanent write protection. The prmwp bit shall be set to zero by the device server when the volume is demounted or when a volume is mounted with permanent write protection disabled. The prmwp shall be set to one by the device server when a volume is mounted with permanent write protection enabled. These changes of state shall not cause a unit attention condition. If the application client sets the prmwp bit to one while no volume is mounted, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to NOT READY. The additional sense information shall be set to LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED. If the application client sets the prmwp bit to one when the logical position is not at BOP 0, the device server shall return CHECK CONDITION status. The sense key shall be set

to ILLEGAL REQUEST. The additional sense information shall be set to POSITION PAST BEGINNING OF MEDIUM. If the application client attempts to change the prmwp bit from one to zero, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to DATA PROTECT. The additional sense information shall be set to PERMANENT WRITE PROTECT. If the device configuration page is savable, the prmwp bit shall be saved as zero, regardless of the current setting.

5.4.3.3 Medium partition page(1)

The medium partition page(1) (see Table 47) is used to specify the first group of medium partitions. Additional groups are specified in medium partition pages(2-4). Fields indicating the current state of the partitions for the medium on any of the medium partition pages(1-4) shall be changed by the device server to the current medium state on a not ready to ready transition when the medium state changes from demounted to mounted. The physical placement and order of medium partitions are not specified by this standard.

NOTE 33 Since defining partitions may require reformatting the medium for some implementations, an implicit write to the medium may occur as a result of a MODE SELECT command that supplies these parameters.

Table 47 – 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		pofm	clear	addp
5	medium format recognition							
6	Reserved				partition units			
7	Reserved							
Partition size descriptor(s)								
8	(MSB)	partition size						
9								(LSB)
$n-1$	(MSB)	partition size						
n								(LSB)

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 maximum additional partitions field is a logical unit-defined value indicating the maximum number of additional partitions supported by the logical unit. A value of zero returned by the MODE SENSE command 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 the MODE SENSE command shall report one less than the number of partitions on the medium 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 if the pofm bit is set to zero. Setting this bit to one when pofm is set to zero may only be valid at beginning-of-partition and is mutually exclusive with the sdp and idp bits. The partition size descriptors are ignored by the MODE SELECT command when the fdp bit is set to one. The logical unit may assign any number of partitions from 1 to (maximum additional partitions + 1).

NOTE 34 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 if the pofm bit is set to zero. The logical unit shall partition the medium into $n + 1$ partitions numbered 0 through n. Setting this bit to one when pofm is set to zero may only be valid at beginning-of-partition and it is mutually exclusive with the fdp and idp fields. The partition size descriptors are ignored by the MODE SELECT command when the sdp bit is set to one.

An initiator-defined partitions (idp) bit of one indicates that the logical unit shall partition the medium as defined by the additional partitions defined field and the partition size descriptors if the pofm bit is set to zero. Setting this bit to one when pofm is set to zero may only be valid at beginning-of-partition and is mutually exclusive with the fdp and sdp fields The number of non-zero partition size descriptors received in medium partition pages (1-4) shall be one more than the additional partitions defined value. The size of partition 0 shall be non-zero.

A logical unit is not required to retain the method used to partition the medium. The device server shall set only one of the idp, fdp or sdp fields in the MODE SENSE data. If a volume was previously partitioned through a MODE SELECT command with fdp or sdp set to one, a device server may set idp to one in subsequent MODE SENSE data since the volume has been initiator partitioned. However, in a MODE SELECT command, the application client cannot use IDP set to one in place of FDP or SDP set to one.

NOTE 35 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 being 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 48.

Table 48 – PSUM values

Code	Description	Support
00b	bytes (unit of one)	Optional
01b	kilobytes (103 bytes)	Optional
10b	megabytes (106 bytes)	Optional
11b	10 (PARTITION UNITS) bytes	Optional

The partition units field defines the size of the partition size descriptors when the psum field is set to 11b. A value of n in the partition units field shall define the units of the partition size descriptors as 10^n bytes. If the partition units field is supported, all possible values shall be supported. A logical unit is not required to retain the partition units used to partition the medium. If psum is not equal to 11b, the partition units field is undefined. Some values of the partition units field may result in no legal non-zero partition size descriptors.

A partition on format (pofm) bit of one indicates that the MODE SELECT command shall not cause changes to the partition sizes or user data, either recorded or buffered. If pofm is set to one, actual medium partitioning occurs when the device server receives a subsequent FORMAT MEDIUM command (see 5.3.2). When the FORMAT MEDIUM command partitions the medium, it shall do so based on the contents of the mode data for medium partition pages (1-4). If pofm is set to one, field values specified by a MODE SELECT command for all medium partition pages (1-4) shall not be changed by the device server before the medium is unloaded or the device is reset. Some field checking may be performed by the MODE SELECT command. However, there is no guarantee that any subsequent partitioning during a FORMAT MEDIUM command will complete with no errors.

A pofm bit of zero indicates that the MODE SELECT command shall alter the partition information for the medium if any of the sdp, fdp or idp bits are set to one.

A clear bit of zero and an addp bit of zero indicate SCSI-2 compatibility since that standard did not specify any mandatory behavior, the logical unit may logically erase any or all partitions when one of the idp, fdp or sdp fields is set to one by a MODE SELECT command.

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 command partition size descriptor and the current partition size differ, the logical unit shall truncate or extend the partition, whichever is appropriate. If the MODE SELECT command partition size is zero and the current partition size is non-zero, the partition shall be logically removed from the medium, resulting in the loss of all data in that partition. If the MODE SELECT command partition size is equivalent to the current partition size, no change in the partition size shall result. If the logical unit is unable to perform the operation or if such an operation would cause loss of valid data in any partition that exists both before and after the MODE SELECT or FORMAT MEDIUM command, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the addition sense code set to PARAMETER VALUE INVALID. If the addp bit is set to one and either addp is not supported or the fdp field is set to one the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier 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 partition's defined value plus one.

If both the addp and clear fields are set to one, the logical unit shall logically erase all partitions that differ in size from the corresponding partition size descriptor in the MODE SELECT data. Partitions with the same size as the MODE SELECT data size shall retain all existing data. If the logical unit is incapable of supporting the changes requested without loss of data, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier 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 and an additional sense code qualifier set to INVALID FIELD IN PARAMETER LIST.

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

a) the mode select partition size, psum, and partition units fields are exactly the same as those returned by MODE SENSE command;
b) the mode select partition size field value is within plus or minus one of the current size when the current size is converted to the units of the mode select psum or partition units field; or
c) the mode select partition size is FFFFh and the current size would return FFFFh if expressed in the units of the mode select psum or partition units field.

The MEDIUM FORMAT RECOGNITION field indicates the logical unit's capability to automatically identify the medium format and partition information when reading a volume. The value in this field may be different following a medium change. The medium format recognition field values are shown in Table 49.

Table 49 – Medium format recognition values

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 to FFh	Reserved

NOTE 36 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 partition size fields within the partition size descriptor list define the approximate size of the respective partitions in the units specified in the psum and partition units fields. Partitions are numbered by their relative position in the partition size descriptor list, starting at 0. Only partition numbers in the range of 0 to n where n is less than or equal to 63 may have size descriptors in this 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 are defined in medium partition pages (2-4) (see 5.4.3.4). Partitions not assigned shall have a partition size descriptor of 0. The logical unit may support more partitions than partition size descriptors. A logical unit may support more partition size descriptors than supported by the medium. All partition size descriptors representing a partition number greater than the maximum additional partition count shall be 0. The partition size descriptors are undefined if the logical unit is not ready. A MODE SELECT command partition size descriptor of FFFFh requests that the logical unit allocate all remaining partition space to that partition. A MODE SENSE command shall return a partition size descriptor of FFFFh if the partition size, in units of psum or partition units, is greater than or equal to FFFFh. If insufficient space exists on the medium for the requested partition sizes or if multiple partition size descriptors are set to FFFFh, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier set to INVALID FIELD IN PARAMETER LIST. A device server may round, as described by the MODE SELECT command in ANSI INCITS 351-2001, any partition size to the nearest valid partition size.

NOTE 37 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.

5.4.3.4 Medium partition page(2-4)

The medium partition page(2-4) (see Table 50) is used to specify additional groups of medium partitions. The first group is specified in the medium partition page(1) (see 5.4.3.3). Fields indicating the current state of the partitions for the medium on any of the medium partition pages(1-4) shall be changed by the device server to the current medium state when the medium state changes from demounted to mounted.

Table 50 – 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)								
2	(MSB)	partition size						
3								(LSB)
n – 1	(MSB)	partition size						
n								(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 additional page codes defined for partition size definition are 12h, 13h and 14h. Up to a maximum of 64 partitions may be defined in each of these pages. The partition size descriptors are numbered from n*64 + 0 to n*64 + 63 where n is equal to page code hex value minus 11h. The partition size field 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 are specified in the psum and partition units fields of the medium partition page(1) (see 5.4.3.3).

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:

a) the medium partition page defines only partitions that are invalid for the logical unit; or

b) the logical unit does not support idp set to one as defined in the medium partition page(1) (see 5.4.3.3).

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 38 It is recommended that sufficient medium partition pages be supported to include 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.

If the MODE SELECT data 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 psum, partition units, addp and clear values 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 and an additional sense code qualifier set to INVALID FIELD IN PARAMETER LIST; or

b) the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier set to INVALID FIELD IN PARAMETER LIST.

The addp and clear fields do not reflect the state of the medium. These fields depend on settings provided by the application client.

If the MODE SELECT data 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 command. 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. The sense key shall be set to ILLEGAL REQUEST with the additional sense code and an additional sense code qualifier set to INVALID FIELD IN PARAMETER LIST.

If the MODE SELECT data 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,

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; or

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 partition's 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 39 It is strongly suggested that MODE SELECT command either send no medium partition pages or send all supported medium partition pages.

5.4.3.5 Read-write error recovery page

The read-write error recovery page (see Table 51) specifies the error recovery and reporting parameters that the device server shall use when transferring data between the device and the medium. These parameters do not affect protocol-level recovery procedures or positioning error recovery procedures.

Table 51 – 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.								

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.

A transfer block (tb) bit of one indicates that the device server shall use its best effort to transfer a data block that cannot be read successfully within the specified read recovery limits to the application client before CHECK CONDITION status is returned. A tb bit of zero indicates that an unrecoverable data block shall not be transferred to the application client. Data blocks that are recoverable within the recovery limits are always transferred, regardless of the value of the tb bit.

An enable early recovery (eer) bit of one indicates that the logical unit shall use the most expedient error recovery algorithm (for example, 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 (for example, 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 shall 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 or write error occurs. All data from the recovered block shall be transferred prior to terminating the read or 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 field specifies the number of times that the logical unit should attempt its recovery algorithm during a read operation before an unrecoverable error is reported. A read retry count of zero indicates that the logical unit shall not use its recovery algorithm during read operations.

The write retry count field specifies the number of times that the logical unit should attempt its recovery algorithm during a write operation before an unrecoverable error is reported. A write retry count of zero indicates that the logical unit shall not use its recovery algorithm during write operations.

5.4.3.6 Informational exceptions control page

The informational exceptions control page (see Table 52) is used to specify the parameters for the control of TapeAlert specific informational exception conditions for a sequential-access device.

Table 52 – Informational exceptions control page

Bit Byte	7	6	5	4	3	2	1	0
0	ps	Rsvd	page code (1Ch)					
1	page length (0Ah)							
2	perf	Resvd	ebf	EWasc	DExcpt	Test	Rsvd	Logerr
3	Reserved				mrie			
4	(MSB)	interval timer						
:								
7								(LSB)
8	(MSB)	report count/test flag number						
:								
11								(LSB)

Refer to ANSI INCITS 351-2001 for a description of the Logerr bit.

A test (test) bit of one indicates the target shall generate test/false informational exception conditions as follows:

c) if the test bit is one and the test flag number value is zero, the target shall generate a false informational exception condition based on the mrie field (the interval timer field is ignored and the report count field is used as the test flag number). When a false informational exception condition is posted, the TapeAlert flags in the log page shall not be modified. True informational exception conditions shall have priority over false informational exception conditions. The test bit shall be automatically cleared when the false informational exception condition is posted on the first command (excluding INQUIRY and REQUEST SENSE commands) that is received with no real informational exception condition pending. The false informational exception condition shall be reported in the method specified by the mrie value and the ASC/ASCQ set to FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE). If the test and dexcpt bits are one and the test flag number value is zero, the target shall return CHECK CONDITION. The sense key shall be set to Illegal Request and the ASC/ASCQ set to Invalid Field in Parameter List; or

d) if the test bit is one and the test flag number value is a valid non-zero value, the target shall generate or clear a test informational exception condition based on the test flag number value as described in Table 53.

Table 53 – TapeAlert test descriptions

Value	Description
01h to 40h	Set the TapeAlert flag indicated by the test flag number field in the log page. Once the TapeAlert flag is set, it shall be processed normally based on the dexcpt, mrie, interval count and report count values.
–01h to –40h	Clear the TapeAlert flag indicated by the absolute value of the test flag number field in the log page. Clearing the flag in this way is equivalent to performing the specified corrective action for that flag, allowing a true informational exception condition to be set if the true error condition occurs for that flag.
7FFFh	Set all of the supported TapeAlert flags in the log page. Once the supported TapeAlert flags are set, they shall be processed normally based on the dexcpt, mrie, interval count and report count values.

A test bit of zero indicates the target shall not generate any test/false informational exception conditions. The value of the test bit returned by a MODE SENSE command shall be zero. The default value of the test bit is zero.

Refer to ANSI INCITS 351-2001 for a description of the DExcpt bit. The default value of DExcpt is one.

Refer to ANSI INCITS 351-2001 for a description of the EWasc bit. The default value of EWasc is zero.

Refer to ANSI INCITS 351-2001 for a description of the ebf bit. The default value of ebf is zero.

Refer to ANSI INCITS 351-2001 for a description of the perf bit. The default value of perf is zero.

Refer to ANSI INCITS 351-2001 for a description of the mrie field. If the informational exception condition was generated by an event that caused a real CHECK CONDITION to occur, then this real CHECK CONDITION will override (i.e. be used instead of) the CHECK CONDITION defined in mrie modes 01h to 05h. The default mrie value is 03h. For mrie modes 01h to 06h, an ASC/ASCQ of FAILURE PREDICTION THRESHOLD EXCEEDED indicates that a TapeAlert event has occurred on the device. Detailed information about the event is stored in the TapeAlert log page. Note that the mrie modes do not effect the logging of events in the TapeAlert log page.

Refer to ANSI INCITS 351-2001 for a description of the interval timer field. The default interval timer value is zero.

The report count/test flag number field has a dual purpose and is described as follows:

If the test bit is set to zero, the value of the report count/test flag number field represents the report count. The report count value shall be returned in response to a MODE SENSE command. When the report count field is set to zero, this indicates that there is no limit on the number of times the target shall report the informational exception condition. When the report count field is not set to zero, this indicates the number of times to report an informational exception condition. If multiple TapeAlert flags are set simultaneously, the flags are reported as a single informational exception condition. The report count is controlled by the newest flag that is set. The default report count/test flag number value is zero.

If the test bit is set to one, the value of the report count/test flag number field represents the test flag number. The test flag number value shall not be returned in response to a MODE SENSE command. Valid values of test flag number are –40h to 40h and 7FFFh. Negative numbers shall be represented using the two's complement method. If the test flag number is set to an invalid value, the device server shall return CHECK CONDITION. The sense key shall be set to ILLEGAL REQUEST and the ASC/ASCQ set to Invalid Field in Parameter List. If the device server does not support a valid test flag number and the test bit is set to one, the device server shall return CHECK CONDITION. The sense key shall be set to Illegal Request and the ASC/ASCQ set to Invalid Field in Parameter List.

6 Printer devices

6.1 Model for printer devices

The printer command set includes capability for the printer-controlling logical unit which may be functionally separate from the physical printer device (see Figure 11) or may be 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.

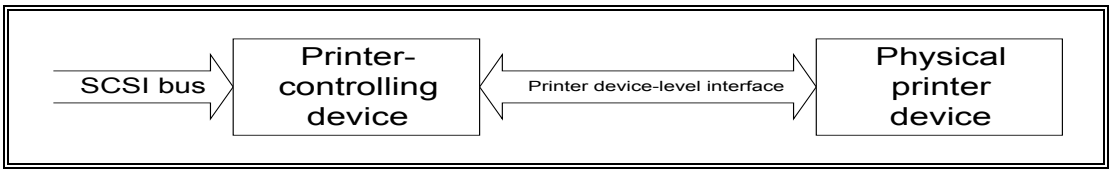

Figure 11 – SCSI printer model

Specific control mechanisms are defined in mode pages for two industry-standard interfaces known as the line printer interface (for example, 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 is 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, that are not convenient to handle in a transparent way, are specified in 6.2.

This International Standard does not specify the character set 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 ANSI INCITS 351-2001) are mandatory for printer devices. Element reservations are not supported by this model.

A printer device is ready when PRINT commands may be executed. A printer is not ready if a print medium is unavailable for use, either from a lack of medium or consumables, or by operator intervention. Such a device, with medium unavailable, normally returns CHECK CONDITION status and sets the sense key to NOT READY and the additional sense code and an additional sense code qualifier shall indicate the reason for being NOT READY.

6.2 Commands for printer devices

6.2.0 General

The commands for printer devices shall be as shown in Table 54.

Table 54 – Commands for printer devices

Command name	Operation code	Type	Reference
Obsolete	40h		3.3.7
COMPARE	39h	O	SPC-2[a]
COPY	18h	O	SPC-2[a]
COPY AND VERIFY	3Ah	O	SPC-2[a]
FORMAT	04h	O	6.2.1
INQUIRY	12h	M	SPC-2[a]
LOG SELECT	4Ch	O	SPC-2[a]
LOG SENSE	4Dh	O	SPC-2[a]
MODE SELECT (6)	15h	M	SPC-2[a]
MODE SELECT (10)	55h	O	SPC-2[a]
MODE SENSE (6)	1Ah	M	SPC-2[a]
MODE SENSE (10)	5Ah	O	SPC-2[a]
PERSISTENT RESERVE IN	5Eh	O	SPC-2[a]
PERSISTENT RESERVE OUT	5Fh	O	SPC-2[a]
PRINT	0Ah	M	6.2.2
READ BUFFER	3Ch	O	SPC-2[a]
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	SPC-2[a]
RECOVER BUFFERED DATA	14h	O	6.2.3
RELEASE(6)	17h	M	SPC-2[a]
RELEASE(10)	57h	M	SPC-2[a]
REPORT LUNS	A0h	M	SPC-2[a]
REQUEST SENSE	03h	M	SPC-2[a]
RESERVE(6)	16h	M	SPC-2[a]
RESERVE(10)	56h	M	SPC-2[a]
SEND DIAGNOSTIC	1Dh	M	SPC-2[a]
SLEW AND PRINT	0Bh	O	6.2.4
STOP PRINT	1Bh	O	6.2.5
SYNCHRONIZE BUFFER	10h	O	6.2.6
TEST UNIT READY	00h	M	SPC-2[a]
WRITE BUFFER	3Bh	O	SPC-2[a]

Key:
M = command implementation is mandatory
O = command implementation is optional

[a] See ANSI INCITS 351-2001.

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. For printer devices, all other operation codes are reserved for future standardization.

6.2.1 FORMAT command

The FORMAT command (see Table 55) provides a means for the application client to specify forms or fonts to printers that support programmable forms or fonts. The format information is vendor-specific.

Table 55 – 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 56.

Table 56 – Format type values

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

The transfer length field specifies the length in bytes of format information that shall be transferred from the initiator. A transfer length of zero indicates that no format information shall be transferred. This condition shall not be considered an error.

6.2.2 PRINT command

The PRINT command (see Table 57) transfers the specified number of bytes from the application client to the device server to be printed.

Table 57 – 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 field specifies the length in bytes of data that shall transferred from the initiator. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered an error.