WORKING
DRAFT

X3T10
Project 996D

Revision 1
Feb-1995

NOTE: This file is provided to facilitate review and comments on the draft SCSI-3 SBC standard. It is NOT intended as a substitute for obtaining a paper copy of this document. Please contact Global Engineering Documents at (800) 854-7179 or (303) 792-2181 to obtain a paper copy.

Information Technology - SCSI-3 Block Commands (SBC)

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

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

ASC X3T10 SBC Technical Editor: Thomas "Rick" Tewell
Sequoia Advanced Technologies, Inc.
55 Shaver Street
San Rafael, CA 94901
USA

Telephone: (415) 459-7978
Facsimile: (415) 459-7988
Email: thomas.tewell@seqadvtech.com

Reference number
ISO/IEC ****** : 199x
ANSI X3.*** - 199x
Printed February 27, 1995 9:00am

POINTS OF CONTACT:

X3T10 Chair
John B. Lohmeyer
Symbios Logic, Inc.
1635 Aeroplaza Drive
Colo Spgs, CO 80916

Telephone: (719) 573-3362
Facsimile: (719) 573-3037
Email: john.lohmeyer@hmpd.com

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

Telephone: (408) 957-7817
Facsimile: (408) 957-7193
Email: ljlamers@aol.com

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

Telephone: (202) 626-5738
Facsimile: (202) 638-4922

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

SCSI Bulletin Board
(719) 574-0424

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

Telephone: (303) 792-2181 or (800) 854-7179
Facsimile: (303) 792-2192

ABSTRACT

This standard specifies the functional requirements for the SCSI-3 Block Command set (SBC). SBC permits SCSI-3 block logical units such as flexible disks, rigid disks, optical disks, etc., to attach to computers and provides the definition for their use.

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

PATENT STATEMENT

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

As of the date of publication of this standard and following calls for the identification of patents that may be required for the implementation of the standard, no such claims have been made. No further patent search is conducted by the developer or the publisher in respect to any standard it processes. No representation is made or implied that licenses are nor required to avoid infringement in the use of this standard.

Revision Information:

The following paragraphs summarize the editing work undertaken for each revision of this draft standard. Revisions are listed in reverse chronological order. This presumes that most reviewers are interested in the recent changes in this draft standard.

Revision 1 (27 Feb 1995)

Changed text to reflect SAM conventions.
Updated table borders.
Added following approved X3T10 changes:

- 90-021R4 - Additional SCSI Caching Control - Revision 4
- 91-014R6 - Control of SCSI Device Power Consumption
- 91-106R4 - Format Status Page Log
- 92-006R1 - Enhancement of Command Sets for Removable Medium Devices with Medium Changers
- 92-130R2 - Synchronization Status and Errors
- 92-217R1 - SCSI Caching Page Changes
- 93-041R2 - SCSI-3 Changes for Dual Port Featurecd \
- 95-???? - Replacement of 93-077 presented at Tahoe meeting by Gene Milligan

Figures added.
Table of Contents valid.
Figure list valid.
Table list valid.
Updated John Lohmeyer's vital statistics.

Revision 0 (01 Nov 1994)

This is the initial revision of the SBC document.

Contents

POINTS OF CONTACT: II

ABSTRACT II

PATENT STATEMENT II

REVISION INFORMATION: III

REVISION 1 (27 FEB 1995) III
REVISION 0 (01 NOV 1994) III

CONTENTS V

FIGURES VIII

TABLES IX

FOREWORD XIII

INTRODUCTION XIV

1 SCOPE 1

2 NORMATIVE REFERENCES 3

2.1 COREQUISITE STANDARDS 3
2.2 REFERENCE STANDARDS 3

3 DEFINITIONS, SYMBOLS AND ABBREVIATIONS 4

3.1 DEFINITIONS SPECIFIC TO DIRECT ACCESS LOGICAL UNITS 4
3.2 DEFINITIONS SPECIFIC TO OPTICAL MEMORY LOGICAL UNITS 4
3.3 DEFINITIONS SPECIFIC TO WRITE-ONCE LOGICAL UNITS 4
3.4 SYMBOLS AND ABBREVIATIONS 5

4 GENERAL 6

4.1 OVERVIEW 6
4.2 CONVENTIONS 6

5 SCSI-3 BLOCK LOGICAL UNIT MODELS 7

5.1 DIRECT-ACCESS LOGICAL UNIT MODEL 7
5.1.1 Removable medium 7
5.1.2 Logical blocks 8
5.1.3 Ready state 8
5.1.4 Power conditions 8
5.1.4 Initialization 9
5.1.5 Medium defects 9
5.1.6 Data cache 10
5.1.7 Reservations 11
5.1.8 Seek and rezero 12
5.1.9 Notched drives 12
5.1.10 Rotational position locking 12
5.1.11 Relative addressing 13
5.1.12 Error reporting 13

5.1.13 Examples ... 14
5.2 MODEL FOR OPTICAL MEMORY LOGICAL UNITS ... 16
5.2.1 Defect management ... 17
5.2.2 Error reporting ... 17
5.3 MODEL FOR WRITE-ONCE LOGICAL UNITS ... 19
5.3.1 Logical blocks ... 19
5.3.2 Initialization ... 19
5.3.3 Physical medium defects ... 19
5.3.4 Error reporting ... 20

6 COMMANDS FOR BLOCK DEVICE LOGICAL UNITS ... 21

6.1 COMMANDS FOR DIRECT-ACCESS LOGICAL UNITS. ... 21
6.1.1 FORMAT UNIT command ... 22
6.1.2 LOCK UNLOCK CACHE command ... 30
6.1.3 PRE-FETCH command ... 31
6.1.4 PREVENT ALLOW MEDIUM REMOVAL command ... 32
6.1.5 READ(6) command ... 33
6.1.6 READ(10) command ... 34
6.1.7 READ CAPACITY command ... 35
6.1.8 READ DEFECT DATA command ... 36
6.1.9 READ LONG command ... 38
6.1.10 REASSIGN BLOCKS command ... 39
6.1.11 RELEASE command ... 41
6.1.12 RESERVE command ... 43
6.1.13 REZERO UNIT command ... 47
6.1.14 SEARCH DATA commands ... 48
6.1.15 SEEK(6) and SEEK(10) commands ... 51
6.1.16 SET LIMITS command ... 52
6.1.17 START STOP UNIT command ... 53
6.1.18 SYNCHRONIZE CACHE command ... 55
6.1.19 VERIFY command ... 56
6.1.20 WRITE(6) command ... 57
6.1.21 WRITE(10) command ... 58
6.1.22 WRITE AND VERIFY command ... 59
6.1.23 WRITE LONG command ... 60
6.1.24 WRITE SAME command ... 61
6.2 COMMANDS FOR OPTICAL MEMORY LOGICAL UNITS ... 62
6.2.1 ERASE(10) command ... 64
6.2.2 ERASE(12) command ... 65
6.2.3 MEDIUM SCAN command ... 66
6.2.4 READ(12) command ... 69
6.2.5 READ DEFECT DATA(12) command ... 70
6.2.6 READ GENERATION command ... 71
6.2.7 READ UPDATED BLOCK(10) command ... 72
6.2.8 SEARCH DATA(12) commands ... 73
6.2.9 SET LIMITS(12) command ... 74
6.2.10 UPDATE BLOCK command ... 75
6.2.11 VERIFY(10) command ... 76
6.2.12 VERIFY(12) command ... 77
6.2.13 WRITE(10) command ... 78
6.2.14 WRITE(12) command ... 79
6.2.15 WRITE AND VERIFY(10) command ... 80
6.2.16 WRITE AND VERIFY(12) command ... 81
6.3 COMMANDS FOR WRITE-ONCE LOGICAL UNITS ... 82

7 PARAMETERS FOR BLOCK DEVICE LOGICAL UNITS 84

7.1 PARAMETERS FOR DIRECT-ACCESS LOGICAL UNITS 84
7.1.1 Diagnostic parameters 84
7.1.2 Log parameters 89
7.1.3 Mode parameters 91
7.2 PARAMETERS FOR OPTICAL MEMORY LOGICAL UNITS 118
7.2.1 Diagnostic parameters 118
7.2.2 Log parameters 118
7.2.3 Mode parameters 118
7.3 PARAMETERS FOR WRITE-ONCE LOGICAL UNITS 121

Figures

FIGURE 1-SCSI-3 DOCUMENT ROAD MAP 1

FIGURE 2 - POWER CONDITIONS FLOWCHART 107

Tables

TABLE 1 - COMMANDS FOR DIRECT-ACCESS LOGICAL UNITS 21

TABLE 2 - FORMAT UNIT COMMAND 22

TABLE 3 - FORMAT UNIT PARAMETER LIST 23

TABLE 4 - DEFECT LIST HEADER 24

TABLE 5 - FORMAT UNIT DEFECT DESCRIPTOR FORMAT AND REQUIREMENTS 25

TABLE 6 - DEFECT DESCRIPTOR - BLOCK FORMAT 27

TABLE 7 - DEFECT DESCRIPTOR - BYTES FROM INDEX FORMAT 27

TABLE 8 - DEFECT DESCRIPTOR - PHYSICAL SECTOR FORMAT 28

TABLE 9 - INITIALIZATION PATTERN DESCRIPTOR 28

TABLE 10 - INITIALIZATION PATTERN MODIFIER 29

TABLE 11 - INITIALIZATION PATTERN TYPE 29

TABLE 12 - LOCK UNLOCK CACHE COMMAND 30

TABLE 13 - PRE-FETCH COMMAND 31

TABLE 14 - PREVENT ALLOW MEDIUM REMOVAL COMMAND 32

TABLE 15 - READ(6) COMMAND 33

TABLE 16 - READ(10) COMMAND 34

TABLE 17 - READ CAPACITY COMMAND 35

TABLE 18 - READ CAPACITY DATA 35

TABLE 19 - READ DEFECT DATA COMMAND 36

TABLE 20 - READ DEFECT DATA DEFECT LIST 37

TABLE 21 - READ LONG COMMAND 38

TABLE 22 - REASSIGN BLOCKS COMMAND 39

TABLE 23 - REASSIGN BLOCKS DEFECT LIST 39

TABLE 24 - RELEASE COMMAND 41

TABLE 25 - RESERVE COMMAND 43

TABLE 26 - DATA FORMAT OF EXTENT DESCRIPTORS 44

TABLE 27 - RESERVATION TYPES 45

TABLE 28 - REZERO UNIT COMMAND 47

TABLE 29 - SEARCH DATA COMMANDS 48

TABLE 30 - SEARCH DATA PARAMETER LIST 49

TABLE 31 - SEEK(6) COMMAND 51

TABLE 32 - SEEK(10) COMMAND 51

TABLE 33 - SET LIMITS COMMAND 52

TABLE 34 - START STOP UNIT COMMAND 53

TABLE 35 - POWER CONDITIONS 54

TABLE 36 - SYNCHRONIZE CACHE COMMAND 55

TABLE 37 - VERIFY COMMAND 56

TABLE 38 - WRITE(6) COMMAND 57

TABLE 39 - WRITE(10) COMMAND 58

TABLE 40 - WRITE AND VERIFY COMMAND 59

TABLE 41 - WRITE LONG COMMAND 60

TABLE 42 - WRITE SAME COMMAND 61

TABLE 43- COMMANDS FOR OPTICAL MEMORY LOGICAL UNITS 62

TABLE 44 - ERASE(10) COMMAND 64

TABLE 45 - ERASE(12) COMMAND 65

TABLE 46 - MEDIUM SCAN COMMAND 66

TABLE 47 - MEDIUM SCAN PARAMETER LIST 67

TABLE 48 - READ(12) COMMAND 69

TABLE 49 - READ DEFECT DATA(12) COMMAND 70

TABLE 50 - READ DEFECT DATA(12) LIST HEADER 70

TABLE 51 - READ GENERATION COMMAND 71

TABLE 52 - MAXIMUM GENERATION DATA BLOCK 71

TABLE 53 - READ UPDATED BLOCK(10) COMMAND 72

TABLE 54 - SEARCH DATA(12) COMMANDS 73

TABLE 55 - SET LIMITS(12) COMMAND 74

TABLE 56 - UPDATE BLOCK COMMAND 75

TABLE 57 - VERIFY COMMAND 76

TABLE 58 - VERIFY(12) COMMAND 77

TABLE 59 - WRITE(10) COMMAND 78

TABLE 60 - WRITE(12) COMMAND 79

TABLE 61 - WRITE AND VERIFY(10) COMMAND 80

TABLE 62 - WRITE AND VERIFY(12) COMMAND 81

TABLE 63 - COMMANDS FOR WRITE-ONCE LOGICAL UNITS 82

TABLE 64 - DIAGNOSTIC PAGE CODES 84

TABLE 65 - TRANSLATE ADDRESS PAGE - SEND DIAGNOSTIC 84

TABLE 66 - TRANSLATE ADDRESS PAGE - RECEIVE DIAGNOSTIC 85

TABLE 67 - DRIVE STATUS PAGE - SEND DIAGNOSTIC 86

TABLE 68 - DRIVE STATUS PAGE - RECEIVE DIAGNOSTIC 87

TABLE 69 - SYNCHRONIZATION STATUS 87

TABLE 70 - LOG PAGE CODES 89

TABLE 71 - FORMAT STATUS PAGE 89

TABLE 72 - DIRECT-ACCESS MEDIUM-TYPE CODES 91

TABLE 73 - DEVICE SPECIFIC PARAMETER 92

TABLE 74 - MODE PAGE CODES 92

TABLE 75 - CACHING PAGE 93

TABLE 76 - DEMAND READ AND WRITE RETENTION PRIORITY 94

TABLE 77 - FLEXIBLE DISK PAGE 97

TABLE 78 - EXAMPLES OF TRANSFER RATES 98

TABLE 79 - PIN 34 FIELD 99

TABLE 80 - PIN 4 FIELD 100

TABLE 81 - PIN 34 FIELD 100

TABLE 82 - FORMAT DEVICE PAGE 101

TABLE 83 - REPORTING OF DEFAULT SECTOR FORMATTING SUPPORT 103

TABLE 84 - REPORTING OF CHANGEABLE SECTOR FORMATTING SUPPORT 103

TABLE 85 - MEDIUM TYPES SUPPORTED PAGE 104

TABLE 86 - NOTCH PAGE 105

TABLE 87 - POWER CONDITION PAGE 106

TABLE 88 - READ-WRITE ERROR RECOVERY PAGE 108

TABLE 89 - ERROR RECOVERY BIT DEFINITIONS 109

TABLE 90 - COMBINED ERROR RECOVERY PARAMETER DESCRIPTIONS 110

TABLE 90 - COMBINED ERROR RECOVERY PARAMETER DESCRIPTIONS (CONTINUED) 111

TABLE 90 - COMBINED ERROR RECOVERY PARAMETER DESCRIPTIONS (CONCLUDED) 112

TABLE 91 - RIGID DISK DRIVE GEOMETRY PAGE 114

TABLE 92 - ROTATIONAL POSITION LOCKING 115

TABLE 93 - VERIFY ERROR RECOVERY PAGE 116

TABLE 94 - DIAGNOSTIC PAGE CODES 118

TABLE 95 - LOG PAGE CODES 118

TABLE 96 - OPTICAL MEMORY MEDIUM-TYPE CODES 119

TABLE 97 - OPTICAL MEMORY LOGICAL UNIT SPECIFIC PARAMETER 119

TABLE 98 - OPTICAL MEMORY DENSITY CODES 120

TABLE 99 - MODE PAGE CODES 120

TABLE 100 - OPTICAL MEMORY PAGE 121

Foreword

The SCSI-3 Block Commands (SBC) standard is divided into seven clauses:

Clause 1 is the scope.

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

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

Clause 4 provides an overview of the block logical unit class and the command set. This clause also specifies the conventions used throughout the standard.

Clause 5 describes models for the various categories of block logical units

Clause 6 provides the definitions of all commands unique to block logical units. This clause also provides references to the SPC standard for primary commands used with this logical unit class.

Clause 7 provides the definition of all parameters unique to this logical unit class.

Introduction

The SCSI command set is designed to provide efficient peer-to-peer operation of input/output logical units (disks, tapes, printers, etc.) by an operating system. The SCSI command set assumes an underlying command-response protocol. Action on SCSI commands shall not be deemed completed until a response is received. The response shall include a status that indicates the final disposition of the command.

The SCSI command set provides multiple operating systems concurrent control over one or more input/output logical units. However, the multiple operating systems must properly coordinate their actions or data corruption will result. The SCSI standard provides commands that assist with coordination between multiple operating systems. However, details of the coordination are beyond the scope of the SCSI command set.

This standard defines a logical unit model for SCSI-3 block logical units. Also defined are SCSI commands that may apply to SCSI-3 block logical units. The standard defines the SCSI commands that shall be implemented by SCSI-3 block logical units.

With any technical document there may arise questions of interpretation as new products are implemented. The X3 Committee has established procedures to issue technical opinions concerning the standards developed by the X3 organization. These procedures may result in SCSI Technical Information Bulletins being published by X3.

These Bulletins, while reflecting the opinion of the Technical Committee that developed the standard, are intended solely as supplementary information to other users of the standard. This standard, X3.***-199x, as approved through the publication and voting procedures of the American National Standards Institute, is not altered by these bulletins. Any subsequent revision to this standard may or may not reflect the contents of these Technical Information Bulletins.

Current X3 practice is to make Technical Information Bulletins available through:

Global Engineering	Telephone:	(303)-792-2181 -or-
15 Inverness Way East		(800) 854-7179
Englewood, CO 80112-5704	Facsimile:	(303)-792-2192

1 Scope

This standard defines the command set extensions to facilitate operation of SCSI-3 block logical units. The clause(s) of this standard pertaining to the SCSI-3 block logical unit class, implemented in conjunction with the applicable clauses of SCSI-3 Primary Commands, shall fully specify the standard command set available for SCSI-3 block logical units.

The objectives of the SCSI-3 Block Commands standard is to provide the following:

a) Permits an Application client to communicate with a logical unit that declares itself to be a SCSI-3 block logical unit (peripheral device types of 0-Direct-access device, 4-Write-once device, and 7-Optical memory device) in the device type field of the INQUIRY command response data over a SCSI-3 service delivery subsytem.

b) Define commands unique to the class of SCSI-3 block logical unit.

c) Define control commands to manage the operation of SCSI-3 block logical unit.

d) Define the differences between categories of SCSI-3 block logical units.

Figure 1 is intended to show the relationship of this document to other SCSI-3 standards. The figure is not intended to imply a relationship such as a hierarchy, protocol stack, or system architecture. It indicates the applicability of a standard to the implementation of a given transport.

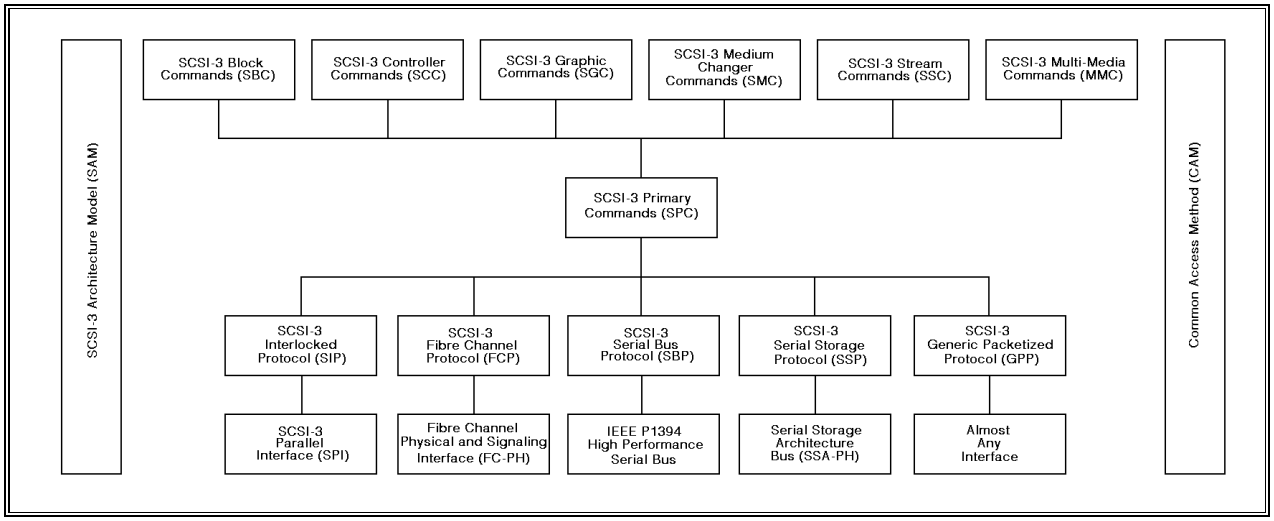

Figure 1-SCSI-3 document road map

The term SCSI is used wherever it is not necessary to distinquish between the versions of SCSI. The Small Computer System Interface-2 (X3.131-1994) is referred to herein as SCSI-2.

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

SCSI-3 Parallel Interface [X3T9.2/855D]
SCSI-3 Interlocked Protocol [X3T9.2/856D]
SCSI-3 Fiber Channel Protocol [X3T9.2/993D]
SCSI-3 Serial Bus Protocol [X3T9.2/992D]
SCSI-3 Generic Packetized Protocol [X3T9.2/991D]
SCSI-3 Architecture Model [X3T9.2/994D]
SCSI-3 Primary Commands [X3T9.2/995D]
SCSI-3 Block Commands [X3T9.2/996D]
SCSI-3 Stream Commands [X3T9.2/997D]
SCSI-3 Graphic Commands [X3T9.2/998D]
SCSI-3 Medium Changer Commands [X3T9.2/999D]
SCSI-3 Controller Commands [X3T9.2/1047D]
SCSI-3 Fast-20 Parallel Interface [X3T9.2/????D]
Multimedia Command Set [X3T9.2/1048D]
Directly-Addressable Device Interface [X3T9.2/964D]
SCSI Common Access Method-2 [X3T9.2/990D]

2 Normative references

2.1 Corequisite standards

The following corequisite standards contain provisions which, through reference in this text, are provisions of this American National Standard. At the time of publication,the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standards are encouraged to investigate the possibility of applying the most recent editions of the following list of standards. Members of IEC and ISO maintain registers of currently valid International Standards. ANSI performs a similar function for American National Standards.

American Standard Code for Information Interchange (ASCII), ANSI X3.4-1977

SCSI-3 Architecture Model, X3T10/994D

SCSI-3 Primary Commands, X3T10/995D

SCSI-3 Medium Changer Commands, X3T10/999D

2.2 Reference standards

Reference standards are listed to help implementers evaluate the complete operating environment for SBC logical units. These standards have no provisions which place requirements on SBC, but they may be useful.

SCSI Common Access Method-2 (CAM-2), X3T10/990-D

3 Definitions, symbols and abbreviations

3.1 Definitions specific to direct access logical units

3.1.1 cache memory: A temporary (and often volatile) data storage area outside the user-accessible area. A cache memory is usually faster to access than the medium and thus has the effect of increasing data throughput by reducing the number of accesses to the medium.

3.1.2 non-volatile medium: Physical storage medium that retains data written to it for a subsequent read operation through a power-on condition. An example of this is a disk device that stores data as magnetic fields that do not require device power to exist.

3.1.3 notch: A notch refers to all or part of the medium having a consistent set of geometry parameters. Notches are used to increase storage capacity by optimizing the number of sectors per track between the inner and outer tracks.

3.1.4 user-accessible: The area of the medium that can be read from or written to by READ and WRITE commands.

3.1.5 volatile medium: Medium that may not retain data written to it for a subsequent read operation through a power-on condition. An example of this is a silicon memory device that loses data written to it if device power is lost.

3.2 Definitions specific to optical memory logical units

3.2.1 blank: The logical block contains no information detectable by the logical unit, or is written with a pattern that appears to the logical unit as no data present. The logical block is considered ready for a write operation.

3.2.2 generation: Indicates a relative revision level of a logical block that has been updated via the UPDATE BLOCK command. A logical block that has never been updated has only one generation associated with it.

3.2.3 read-only medium: This is medium that may not be written by the application client. The media contains data prepared in a manner not defined by this International Standard.

3.2.4 update: To write new data to a logical block without destroying the previous data. After a block has been updated, a normal read returns the most recent generation of the data. Earlier generations are still available after the update.

3.2.5 write-once medium: This is medium that may be written only once by the application client. Logical blocks on write-once media that have not been written are considered blank. Logical blocks on write-once media that have been written may not be written again.

3.3 Definitions specific to write-once logical units

See 3.2.

3.4 Symbols and abbreviations

CDB	command descriptor block
e.g.	for example (illustrative)
i.e.	that is (additional explanation)
I/O	input/output
ID	identifier
LSB	least significant bit
MSB	most significant bit
SCSI	either SCSI-2 or SCSI-3
SCSI-2	the Small Computer System Interface-2 (ANSI X3.131-1994)
SCSI-3	the Small Computer System Interface-3
SPC	SCSI-3 Primary Command Set standard

4 General

4.1 Overview

SCSI-3 logical units which conform to this standard are referred to as SCSI-3 block logical units. This includes the category of logical units commonly referred to as flexible disks, rigid disks, removable rigid disks, erasable optical discs, write once optical discs and read only optical discs (excluding CD-ROM which is defined by the SCSI-3 Multimedia Command Set (MMC) standard).

The common attribute of block logical units is that they are block addressable. That is to say that data are addressed on the logical unit in groups referred to as logical blocks. The number of bytes of data contained in a single logical block is known as the block length. The block length is almost always greater than one byte and is typically (but by no means always) a multiple of 512 bytes. In addition, a logical block does not necessarily bear any relation to the physical block size of the storage medium.

Each logical block has a block length associated with it. This means that the block length for the medium can change from logical block to logical block. However, by convention the block length usually remains constant over the entire capacity of the medium.

This specification is intended to be used in conjunction with the SCSI-3 Architecture Model (SAM) and the SCSI-3 Primary Command Set (SPC) standards.

4.2 Conventions

Certain words and terms used in this International Standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in clause 3 or in the text where they first appear.

Words have the normal technical English definition unless the word or phrase is defined in context or in the glossary, then that definition is used.

Numbered items in SBC do not represent any priority. Any priority is explicitly indicated.

Fields containing only one bit are usually referred to as the name bit instead of the name field.

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

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

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

In all of the text and tables of this standard, the most significant bit of a binary quantity is shown on the left side and represents the highest algebraic value position in the quantity.

The word "shall" is used to indicate a mandatory requirement. If such a requirement is not followed, the results are unpredictable unless indicated otherwise. In addition, if such a requirement is not followed, the implementation is not in conformance with this standard. If a field is specified as not meaningful or it is to be ignored, the entity that receives the field shall not check that field.

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

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

5 SCSI-3 block logical unit models

5.1 Direct-access logical unit model

Direct-access logical units store blocks of data for later retrieval. Each block of data is stored at a unique logical block address. An application client issues WRITE commands to store the blocks of data (write operations) and READ commands to retrieve the blocks of data (read operations). Other commands issued by the application client may also cause write and read operations to occur. A write operation causes a block of data to be written on the medium. A read operation causes a block of data to be read from the medium. A verify operation confirms that a block of data can be read without error from the medium.

Blocks of data are stored by a process that causes localized changes or transitions within the medium. The changes made to the medium to store the blocks of data may be volatile (i.e. not retained through a power cycle) or non-volatile (retained through a power cycle). The medium may be divided in parts that are used for data blocks, parts that are reserved for defect management, and parts that are reserved for use by the controller for the management of the logical unit.

5.1.1 Removable medium

The medium may be removable (e.g. used in a floppy disk drive) or non-removable (e.g. used in a hard disk drive). Removable medium is contained within a cartridge (or jacket) to prevent damage to the recording surfaces. The combination of medium and cartridge is often called a volume.

A volume has an attribute of being mounted or de-mounted on a suitable transport mechanism. A volume is mounted when the direct-access logical unit is capable of performing write or read operations to the medium. A mounted volume may not be accessible by an application client if it is reserved by another application client. A volume is de-mounted at any other time (e.g. during loading, unloading, or storage).

An application client may check whether a volume is mounted by issuing a TEST UNIT READY command. A volume that is loaded may need a START STOP UNIT command issued to become accessible for write or read operations.

The PREVENT ALLOW MEDIUM REMOVAL command allows an application client to restrict the demounting of the volume. This is useful in maintaining system integrity. If the direct-access logical unit implements cache memory, it must ensure that all logical blocks of the medium contain the most recent data prior to permitting demounting of the volume. If the application client issues a START STOP UNIT command to eject the cartridge, and the direct-access logical unit is prevented from demounting by the PREVENT ALLOW MEDIUM REMOVAL command, the START STOP unit command is rejected by the device server.

5.1.1.1 Removable medium with an Attached Medium Changer

When a logical unit is served by a medium changer, control over a media transport element may be effected using media changer commands sent to the logical unit.

The logical unit indicates its ability to support these commands by setting the "MChngr" bit to one in its standard INQUIRY data.

An "MChngr" bit of one indicates that the MOVE MEDIA and READ ELEMENT STATUS commands are supported.

Only one medium transport element is permitted (element 0) and only one data transfer element is permitted.

5.1.2 Logical blocks

Blocks of data are stored on the medium along with additional information that the controller uses to manage the storage and retrieval. The format of the additional information is unique and is hidden from the application client during normal read or write operations. This additional information is often used to identify the physical location of the blocks of data and the address of the logical block, and to provide protection against the loss of the user data.

The address of the first logical block is zero. The address of the last logical block is [n-1], where [n] is the number of logical blocks available on the medium. A READ CAPACITY command may be issued to determine the value of [n-1]. If a command is issued that requests access to a logical block not within the capacity of the medium, the command is terminated with CHECK CONDITION.

The number of bytes of data contained in a logical block is known as the block length. Each logical block has a block length associated with it. The block length may be different for each logical block on the medium. However, in a typical logical unit only one block length is used at a time. The block descriptor in the MODE SENSE data describes the block lengths that are used on the medium. A MODE SELECT command can be used to set up extents. An extent is a specified number of logical blocks that have the specified block length. The FORMAT UNIT command is typically required to change the block length of logical units that support variable block lengths and make the extents that were setup active.

The location of a logical block on the medium does not have a relationship to the location of any other logical block. However, in a typical logical unit the logical blocks are located in an ascending order. The time to access the logical block at address [x] and then the logical block at address [x+1] need not be less than time to access [x] and then [x+100]. The READ CAPACITY with a PMI bit of one is useful in determining where longer access times occur.

5.1.3 Ready state

A direct-access logical unit is ready when medium access commands can be executed. A logical unit using removable media is usually not ready until a volume is mounted. Such a logical unit, with a volume not mounted, normally returns CHECK CONDITION status and sets the sense key to NOT READY.

Some direct-access logical units may be switched from being ready to being not ready by using the START STOP UNIT command. An application client may need to issue a START UNIT command to bring a logical unit ready.

5.1.4 Power conditions

The optional power conditions permit the application client to modify the behavior of a logical unit in a manner which reduces the power required to operate. There is no notification to the application client that a logical unit has entered into one of the power conditions. The power conditions may be controlled by the START STOP UNIT command or the Power Condition Page of the MODE SELECT command. If both methods are being used on the same target/logical unit combination then any START STOP UNIT commands power condition request will override the Power Condition Pages power control. See the START STOP UNIT command description and the MODE SELECT Power Conditions page description for more information.

No power condition shall affect the supply of termination power to the SCSI bus.

The lowest power consumption occurs in the Sleep condition. When in the Sleep condition a logical unit needs a bus reset to be activated.

In the Standby condition a logical unit is capable of accepting commands, but media is not immediately accessible (eg spindle is stopped).

In the Idle condition a logical unit is capable of responding quickly to media access requests. However, a logical unit in the Idle condition may take longer to complete the execution of a command because it may have to activate some circuitry.

In the Active condition a logical unit is capable of responding immediately to media access requests, and operations complete execution in the shortest possible time.

Logical units that contain cache memory shall implicitly perform a SYNCHRONIZE CACHE command for the entire medium prior to entering into any power condition which prevents access the media (eg the spindle being stopped).

The logical unit shall use the Power Condition Page to control the power conditions after a power on or a hard reset until a Start/Stop Unit command is received with the Power Conditions field set to a value other than 0h or 7h.

5.1.4 Initialization

Many direct-access logical units require initialization prior to write or read operations. This initialization is usually performed by a FORMAT UNIT command. Parameters related to the geometry and performance characteristics can be set with the MODE SELECT command prior to the format operation. Some logical units are initialized by means not specified in this International Standard. The time at which this occurs is specific to the implementation of the direct-access logical unit.

Logical units using a non-volatile medium typically save the parameters and only need to be initialized once. However, some mode parameters may need to be initialized after each power-on and reset. A catastrophic failure of the direct-access logical unit may require the FORMAT UNIT command to be reissued.

Logical units that use a volatile medium may need to be initialized at each power-on prior to the execution of read or write operations. Mode parameters may also need initialization.

5.1.5 Medium defects

Any medium has the potential for defects that can cause user data to be lost. Therefore, each logical block may contain information that allows the detection of changes to the user data caused by defects in the medium or other phenomena, and may also allow the data to be reconstructed following the detection of such a change. Some logical units provide the application client control through use of the mode parameters. Some logical units allow the application client to examine and modify the additional information by using the READ LONG and WRITE LONG commands. Some media having a very low probability of defects may not require these structures.

Defects may also be detected and managed during execution of the FORMAT UNIT command. The FORMAT UNIT command defines four sources of defect information. These defects may be reassigned or avoided during the initialization process so that they do not appear in a logical block.

Defects may also be avoided after initialization. The application client issues a REASSIGN BLOCKS command to request that the specified logical block address be reassigned to a different part of the medium. This operation can be repeated if a new defect appears at a later time. The total number of defects that may be handled in this manner can be specified in the mode parameters.

Defect management on direct-access logical units is usually vendor-specific. Logical units not using a removable medium typically optimize the defect management for capacity or performance or both. Logical units that use a removable medium typically do not support defect management (e.g. some floppy disk drives) or use defect management that is based on the ability to interchange the medium.

5.1.6 Data cache

Some direct-access logical units implement cache memory. A cache memory is usually an area of temporary storage in the direct-access logical unit with a fast access time that is used to enhance performance. It exists separately from the blocks of data stored and is normally not directly accessible by the application client. Use of cache memory for write or read operations typically reduces the access time to a logical block and can increase the overall data throughput.

During read operations, the direct-access logical unit uses the cache memory to store blocks of data that the application client may request at some future time.The algorithm used to manage the cache memory is not part of this International Standard. However, parameters are provided to advise the device server about future requests, or to restrict the use of cache memory for a particular request.

During write operations, the direct-access logical unit uses the cache memory to store data that is written to the medium at a later time. This is called a write-back caching algorithm. Thus the command may complete prior to blocks of data being written to the medium. As a result of using a write-back caching algorithm there is a period of time when the data may be lost if a power or a hardware failure occurs. There is also the possibility of an error occurring during the write operation. If an error occurred during the write, it may be reported as a deferred error on a later command. However, the application client can request write-through caching to prevent these circumstances from arising.

When the cache memory fills up with blocks of data that are being kept for possible future access, new blocks of data that are to be kept must replace those currently in cache memory. The disable page out (DPO) bit is used to control replacement of logical blocks in the cache. For write operations, setting this bit to one advises the device server to not replace existing blocks in the cache memory with the write data. For read operations, setting this bit to one causes blocks of data that are being read to not replace existing ones in the cache memory.

Sometimes the application client may wish to have the blocks of data read from the medium instead of from the cache memory. The force unit access (FUA) bit is used to indicate that the device server shall access the physical medium. For a write operation, setting FUA to one causes the device server to complete the data write to the physical medium before completing the command. For a read operation, setting FUA to one causes the logical blocks to be retrieved from the physical medium.

When the DPO and FUA bits are both set to one, write and read operations bypass the cache memory.

When a VERIFY command is executed, a forced unit access is implied, since the blocks of data stored on the medium are being verified. Furthermore,a SYNCHRONIZE CACHE operation (see below) is also implied to write unwritten blocks of data still in the cache memory. These blocks of data must be stored on the medium before the verify operation can begin. The DPO bit is provided since the VERIFY command may cause the replacement of blocks in the cache. The above also applies to the WRITE AND VERIFY command.

Commands may be implemented by the device server that allow the application client to control other behaviour of the cache memory:

LOCK UNLOCK CACHE controls whether certain logical blocks will be held in the data cache for future use. Locking a logical block prevents its replacement by a future access. Unlocking a logical block exposes it to possible replacement by a future access (see 6.1.2).

PRE-FETCH causes a set of logical blocks requested by the application client to be read into the data cache for possible future access. The blocks fetched are subject to later replacement unless they are locked (see 6.1.3).

SYNCHRONIZE CACHE forces any pending write data in the requested set of logical blocks to be stored in the physical medium. This command can be used to ensure that the data was written and any errors reported (see 6.1.18).

The MODE SELECT command defines a page for the control of cache behavior and handles certain basic elements of cache replacement algorithms (see 7.1.3.1).

5.1.7 Reservations

The access enabled or access disabled condition determines when an application client may store or retrieve user data on all or part of the medium. Access may be restricted for read operations, write operations, or both. This attribute may be controlled by an external mechanism or by the RESERVE and RELEASE commands (see 6.1.12 and 6.1.11).

The RESERVE and RELEASE commands define how different types of restricted access may be achieved, and to whom the access is restricted. This subclause describes the interaction of the application client that requested the reservation, and the other application clients.

An application client uses reservations to gain a level of exclusivity in access to all or part of the medium for itself or another application client. It is expected that the reservation will be retained until released. The device server must ensure that the application client with the reservation is able to access the reserved media within the operating parameters established by that application client.

The following paragraphs explain, on a command by command basis, the appropriate device server response when a reservation exists. Unless otherwise noted, the appropriate response to an application client that issues a command to a device server that is reserved to another application client is RESERVATION CONFLICT status.

The CHANGE DEFINITION command is dealt with as follows. If any application client has an extent reservation on a device server, no other application client may affect the operating definition of that application client by use of this command. If the device server allows different operating definitions for each application client, then there is no conflict; otherwise, a reservation conflict occurs.

The COMPARE, COPY, and COPY AND VERIFY commands are evaluated for reservation conflict as if they were normal write and read operations even when a device server is requested to copy to or from itself. For example, if a COPY is issued to logical unit 0 that requests the device server to copy from logical unit 0 to logical unit 1, access to logical unit 1 must also be evaluated for conflict.

The FORMAT UNIT, PREVENT ALLOW MEDIUM REMOVAL (with a prevent bit of one),REZERO UNIT, and START STOP UNIT commands return a RESERVATION CONFLICT status if any other application client has an extent reservation on a device server.

The INQUIRY and REQUEST SENSE commands are not affected by any kind of reservation.

The LOG SELECT, LOG SENSE, MODE SENSE, TEST UNIT READY, READ CAPACITY (PMI set to zero), READ BUFFER, WRITE BUFFER, and READ DEFECT DATA commands are not affected by extent reservations.

The SEEK, LOCK UNLOCK CACHE, PRE-FETCH, and SYNCHRONIZE CACHE commands are evaluated for reservation conflict as if they were normal write or read operations.

The MODE SELECT command is dealt with as follows. If an application client has an extent reservation on a device server, and another application client attempts one of these commands, a reservation conflict occurs if the command affects the manner in which access of the extent by the first application client is performed. If the command does not affect access to the extent, or parameters are saved for each application client, then a conflict does not occur.

The SEND DIAGNOSTIC, RECEIVE DIAGNOSTIC RESULTS commands conflict with an extent reservation only if they affect access to the extent (as with MODE SELECT).

The REASSIGN BLOCKS command may not reassign a block that is in an extent reserved to another application client.

The SET LIMITS command generates a reservation conflict if the logical blocks specified are within an extent reserved to another application client.

ALL other commands are that request read or write operations are evaluated for reservation conflict as described in the RESERVE command.

When a system is integrated with more than one application client, there must be agreement between the application clients as to how media is reserved and released during operations, otherwise, an application client may be locked out of access to a logical unit in the middle of an operation. For example, application client 'A' has a write operation in progress to a direct-access logical unit which has data stored in cache memory. Then, application client 'B' issues a RESERVE command to the direct-access logical unit. As a result, application client 'A' is locked out of issuing a SYNCHRONIZE CACHE command to ensure the integrity of the data. To prevent this from happening, application client 'A' should issue a RESERVE prior to the write command.

5.1.8 Seek and rezero

The SEEK command provides a way for the application client to position the logical unit in preparation for access to a particular logical block at some later time. Since this positioning action is implicit in other commands, the SEEK command may not be useful with some direct-access logical units.

The REZERO UNIT command is provided to bring the direct-access logical unit to a known condition. This International Standard does not specify the condition. The REZERO UNIT command is used in some logical units to position the actuator at cylinder zero. Some logical units return GOOD status without attempting any action.

5.1.9 Notched drives

A notched (also called partitioned or zoned) drive has areas of the medium in which the drive geometry changes. In the simplest case, the entire medium consists of a single notch. Multiple notches are often used to increase capacity of the drive. The notch page is used to indicate the notch for assignment of values to the parameters in the format device page. By sequencing the notch page through each notch, the format device parameters of each notch are set. This is usually done prior to initialization by the FORMAT UNIT command.

5.1.10 Rotational position locking

Rotational position locking is an optional feature implemented in some direct-access logical units to allow the synchronization of spindles between a number of logical units. The rotational position offset feature allows logical units to synchronize spindles at offsets from index. This may be useful in improving performance in systems that implement arrays of logical units.

5.1.11 Relative addressing

Relative addressing is a technique useful in accessing structured data in a uniform manner. Relative addressing is only allowed when commands are linked. An example of relative addressing and linking for SEARCH DATA commands appropriate to direct-access logical units is given in 7.4.3.

The SET LIMITS command is provided to define the limits of a linked chain of relative addressing commands. This gives an additional protection against exceeding a particular set of blocks. The SET LIMITS command has no effect on any other application client.

5.1.12 Error reporting

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

Condition	Sense key
Invalid logical block address	ILLEGAL REQUEST
Unsupported option requested	ILLEGAL REQUEST
Logical unit reset or medium change since last command from this application client	UNIT ATTENTION
Self diagnostic failed	HARDWARE ERROR
Unrecovered read error	MEDIUM ERROR or HARDWARE ERROR
Recovered read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED COMMAND
Attempt to write on write protected medium	DATA PROTECT

In the case of an invalid logical block address, the sense data information field shall be set to the logical block address of the first invalid address.

In the case of an attempt to read a blank or previously unwritten block, the information field shall be set to the logical block address of the first blank block encountered. The data read up to that block shall be transferred (optical memory and write-once logical units only).

In the case of an attempt to write a previously written block when blank checking is enabled, the information field shall be set to the logical block address of the first non-blank block encountered (optical memory and write-once logical units only).

5.1.13 Examples

The following examples show some typical variations of the direct-access logical unit. Other variations are possible.

5.1.13.1 Rotating media

The typical application of a direct-access logical unit is a disk drive. The medium is a disk coated with a material in which flux changes may be induced. The disk drive allows direct and random access to the medium. This is done with an actuator that positions the read-write head, and a rotating disk. Data is stored and retrieved through the interaction of the read-write head and the disk.

The disk is typically divided into cylinders. Each cylinder is typically divided into tracks. Each track is typically divided into sectors. A cylinder is a set of tracks that can be accessed without movement of the actuator. A track is a recording path over which the read-write head travels during one rotation of the disk. A sector is a part of a track that contains the stored data blocks.

A logical block is stored in one or more sectors, or a sector may store more than one logical block. A sector is typically made up of a header, data and a trailer. The header contains a preamble used to synchronize read circuits to the data, an address field to identify the sector, flags to use for defect management, and a checksum that validates the header. The data contains the block of data. The trailer contains checksum or error correction information. The checksum or the error correction information allows the correction of data for medium defects.

A disk drive is ready when the disks are rotating at the correct speed and the read-write circuitry is powered and ready to access the disks. Some disks, particularly removable disks, require the user to issue load or start commands to bring the disk drive to the ready state.

A disk drive will typically have to be formatted prior to the initial access. Exceptions to this are drives that are formatted at the factory and some optical drives with pre-formatted media (see 5.3). A disk drive format will typically create the headers for each sector and initialize the data field. The MODE SELECT command is often used at format time to establish the geometry (number of heads and tracks, sectors per track, etc.) and defect management scheme. Disk drives are usually non-volatile.

The defect management scheme of a disk drive is often shielded from the user, though some aspects can be evaluated and controlled by the application client. The device server will usually reserve some sectors and tracks for recording defect lists and for reassigning defective blocks. The READ LONG and WRITE LONG commands will typically access the user data and checksum portions of the data field so that defects may be induced by the application client to test the defect detection logic of the device server.

Notches find their most typical use in a rotating disk drive. On a disk, the inner tracks are physically shorter than the outer tracks. As a result, if each track is made to store the same number of data bits, the data is packed more densely on the inner tracks than the outer tracks. By using notches, the outer tracks may be made to contain a different number of sectors than the inner tracks, while balancing the data density. This results in increased capacity.

5.1.13.2 Sequential media

Some tape logical units are implemented as a direct access logical unit so that they can be used in disk oriented operating system environments. These logical units are sometimes referred to as random access tape or floppy tape. These logical units might be thought of as a disk drive with one or more long tracks. Access time to a logical block is usually longer than for a disk drive, since the tape must be fast forwarded or rewound to the block. As a result, the SEEK command will often be more useful for a tape than for a disk. The only way an application client may determine if a direct-access logical unit is a tape is by using the medium type code returned by the MODE SENSE command.

5.1.13.3 Memory media

Memory media includes logical units that are traditionally used for primary storage within computer systems, such as solid state static or dynamic random access memories (SRAM or DRAM), or magnetic core or bubble memory.

These logical units are typically non-mechanical, and therefore the entire physical medium may be accessed in virtually the same access time. The data is typically accessed as a bit or byte and this also speeds access time. Memory logical units typically store less data than disks or tapes, and are usually volatile when not protected by battery backup.

5.2 Model for optical memory logical units

An optical memory logical unit is a logical unit that can potentially support a variety of optical media, (e.g. read-only, write-once, erasable, or reversible). In several respects, an optical memory logical unit is similar to a direct-access logical unit. However, optical memory logical units typically offer features that are not available with other logical units, including very large capacity removable media.

These logical units often require the functions that are not found in direct-access logical units such as logical block update, pre-erasure before writing, or scanning for blank medium and twelve-byte command descriptor blocks. This standard has specific device types for write-once and CD-ROM logical units that also use optical media, but are not capable of supporting several types of optical media. A logical unit that uses write-once media can be an optical memory logical unit. Logical units that use read-only media can be optical memory logical units; however, logical units using CD-ROM media have certain unique characteristics and should not be implemented as optical memory logical units.

A model of optical memory logical units is complicated by the nature of one of its potential advantages, that it can support media which has different characteristics. There are three types of optical media in general use, read-only, write-once and reversible. Read-only media are typically used for publishing applications requiring dissemination of large amounts of data, since the data can be replicated on a disk at low cost. Write-once media are used in applications that have large backup or archiving requirements. It is also used in applications that need large amounts of on-line reference information. Reversible media is typically used in applications that need large amounts of temporary storage (e.g. a graphics workstation), and can take advantage of removable media. In some applications, reversible media drives are used in place of direct-access logical units.

Reversible media usually need to be reversed (erased, blanked) before new data can be written. In such cases an erase operation is required before data can be written. Some optical memory logical units perform this erase operation implicit with each write operation that typically impacts the data throughput. Some logical units can perform the erase separately. The ERASE command may be used to erase areas of the medium with a corresponding increase in data throughput on subsequent write operations. Products using optical media should not be implemented as direct-access logical units, due to the overhead penalty on performance from the emulation and the lack of support in direct-access logical units to take advantage of the sophisticated features available with optical memory logical units.

The type of medium supported by the logical unit and the type of medium currently loaded can be determined by examining the MODE SENSE data. One unique feature of optical memory logical units is support of media with mixed types (e.g. media with read-only and write-once areas). The INQUIRY command informs the application client that the logical unit is an optical memory logical unit; the application client should then determine the medium type from the MODE SENSE data. The application client needs to be cognizant of medium type since the logical unit's characteristics can change when the media are changed.

Write-once media can have valid data written to a logical block once. This is an important feature where audit trails and permanent archives are needed. Many optical memory logical units supporting write-once media have the ability to update a logical block, preserving the previous generation of data. These logical units usually provide a means to recover the previous data through use of commands that allow read access to the different generations of data that are stored at the same logical block address.

An important requirement in dealing with optical media is determining which logical blocks contain written data and which are blank. A blank logical block is one that is properly initialized so that data written to it can be recovered. The logical blocks usually have a flag associated with each that indicates whether they have been written or not.

Many of the strategies used to manage write once and erasable media depend on being able to determine the boundary between written and blank areas of the medium. The MEDIUM SCAN command is useful in finding blank areas for subsequent write operations.

5.2.1 Defect management

Defect management can be performed on logical blocks by updating in a manner similar to that used by direct-access logical units with the REASSIGN BLOCKS command. The advantage of using the updating (which is not supported by direct-access logical units) is access to the previous data.

The update operation assigns an alternate physical block to the logical block while simultaneously writing the data to the block. Commands are provided to allow the recovery of previous generations of updated blocks.

Defect management on optical-memory logical units is usually vendor-specific. However there are standards for some types of optical-memory media that specify defect management techniques. These standards, where they exist, may supersede the implementation requirements pertaining to error and defect reporting in this International Standard.

5.2.2 Error reporting

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

Condition	**Sense Key**
Invalid logical block address	ILLEGAL REQUEST
Unsupported option requested	ILLEGAL REQUEST
Logical unit reset or medium change since last command from this application client	UNIT ATTENTION
Self diagnostic failed	HARDWARE ERROR
Unrecovered read error	MEDIUM ERROR or HARDWARE ERROR
Recovered read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED command
Attempt to write on write protected medium	DATA PROTECT
Attempt to read a blank or previously unwritten block	BLANK CHECK
Attempt to write a previously written block and blank checking is enabled	BLANK CHECK
Attempt to write on read-only medium	DATA ROTECT

In the case of an invalid logical block address, the sense data nformation field shall be set to the logical block address of the irst invalid address.

In the case of an attempt to read a blank or previously unwritten block, the sense data information field shall be set to the logical block address of the first blank block encountered. The data read up to that block shall be transferred.

In the case of an attempt to write a previously written block when blank checking is enabled, the sense information field shall be set to the logical block address of the first non-blank block encountered.

5.3 Model for write-once logical units

The model for the write-once logical unit is a variation on the optical memory model. Most of the aspects of a write-once logical unit are similar to optical memory logical units. The differences are summarized in this clause.

5.3.1 Logical blocks

Data may be written to a logical block only once. A subsequent write to a logical block already written may or may not be corrupted, depending on the implementation. Write-once physical media is non-volatile.

The updating of blocks (see 5.2) is discouraged. SCSI write-once logical units are intended to be archival in nature. Data at a logical block address is not expected to change once it is written. The update commands are not defined for this device type. Logical units are not precluded from implementing the update function by this standard, though logical units that require the update function should use the optical memory device type.

Logical units may be able to determine the state of a logical block prior to access. These logical units can determine whether a block is blank or written. This is useful in detecting previously written blocks, and preventing a destructive overwrite. This is also useful in finding blank areas for later writing. The MEDIUM SCAN command can be used to find blank and written areas prior to WRITE and READ access.

5.3.2 Initialization

The FORMAT UNIT command is not defined for write-once logical units. Write-once media is shipped pre-formatted by the manufacturer and is ready for use when mounted.

5.3.3 Physical medium defects

The raw defect rate is typically higher for optical medium than for magnetic medium. Data is usually recovered through the use of sophisticated error correction algorithms. The level of error correction used for data recovery is often can be selected. However, many write-once logical units have a minimum level that is always used and cannot be de-selected through the error recovery mode parameter. Control of the error correction algorithms and level of correction is vendor-specific.

Defect management on write-once logical units is usually vendor-specific. However, there are standards for some types of write-once media that specify defect management techniques. These standards, where they exist, may supersede the implementation requirements pertaining to error and defect reporting in this International Standard.

5.3.4 Error reporting

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

Condition	Sense Key
Invalid logical block address	ILLEGAL REQUEST
Unsupported option requested	ILLEGAL REQUEST
Logical unit reset or medium change since last command from this application client	UNIT ATTENTION
Self diagnostic failed	HARDWARE ERROR
Unrecovered read error	MEDIUM ERROR or HARDWARE ERROR
Recovered read error	RECOVERED ERROR
Overrun or other error that might be resolved by repeating the command	ABORTED command
Attempt to write on write protected medium	DATA PROTECT
Attempt to read a blank or previously unwritten block	BLANK CHECK
Attempt to write a previously written block and blank checking is enabled	BLANK CHECK

In the case of an invalid logical block address, the sense data information field shall be set to the logical block address of the first invalid address.

In the case of an attempt to read a blank or previously unwritten block, the sense data information field shall be set to the logical block address of the first blank block encountered. The data read up to that block shall be transferred.

In the case of an attempt to write a previously written block and blank checking is enabled, the sense information field shall be set to the logical block address of the first non-blank block encountered.

6 Commands for block device logical units

6.1 Commands for direct-access logical units.

The commands for direct-access logical units shall be as shown in table 1.

Table 1 - Commands for direct-access logical units

Command name	Operation code	Type	Subclause
CHANGE DEFINITION	40h	O	SPC
COMPARE	39h	O	SPC
COPY	18h	O	SPC
COPY AND VERIFY	3Ah	O	SPC
FORMAT UNIT	04h	M	6.1.1
INQUIRY	12h	M	SPC
LOCK-UNLOCK CACHE	36h	O	6.1.2
LOG SELECT	4Ch	O	SPC
LOG SENSE	4Dh	O	SPC
MODE SELECT(6)	15h	O	SPC
MODE SELECT(10)	55h	O	SPC
MODE SENSE(6)	1Ah	O	SPC
MODE SENSE(10)	5Ah	O	SPC
MOVE MEDIUM	A5h	O	SMC
PORT STATUS	11h	M[1]	SPC
PRE-FETCH	34h	O	6.1.3
PREVENT-ALLOW MEDIUM REMOVAL	1Eh	O	6.1.4
READ(6)	08h	M	6.1.5
READ(10)	28h	M	6.1.6
READ BUFFER	3Ch	O	SPC
READ CAPACITY	25h	M	6.1.7
READ DEFECT DATA	37h	O	6.1.8
READ ELEMENT STATUS	B8h	O	SMC
READ LONG	3Eh	O	6.1.9
REASSIGN BLOCKS	07h	O	6.1.10
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	SPC
RELEASE	17h	M	6.1.11
REQUEST SENSE	03h	M	SPC
RESERVE	16h	M	6.1.12
REZERO UNIT	01h	O	6.1.13
SEARCH DATA EQUAL	31h	O	6.1.14
SEARCH DATA HIGH	30h	O	6.1.14
SEARCH DATA LOW	32h	O	6.1.14
SEEK(6)	0Bh	O	6.1.15
SEEK(10)	2Bh	O	6.1.15
SEND DIAGNOSTIC	1Dh	M	SPC
SET LIMITS	33h	O	6.1.16
START STOP UNIT	1Bh	O	6.1.17
SYNCHRONIZE CACHE	35h	O	6.1.18
TEST UNIT READY	00h	M	SPC
VERIFY	2Fh	O	6.1.19
WRITE(6)	0Ah	O	6.1.20
WRITE(10)	2Ah	O	6.1.21
WRITE AND VERIFY	2Eh	O	6.1.22
WRITE BUFFER	3Bh	O	SPC
WRITE LONG	3Fh	O	6.1.23
WRITE SAME	41h	O	6.1.24

Key: M = Command implementation is mandatory.
O = Command implementation is optional.
SPC = SCSI-3 Primary Commands
SMC = SCSI-3 Medium Changer Command Set

Notes: [1] Mandatory only if the dual port option is implemented.

The following operation codes are vendor-specific: 02h, 05h, 06h, 09h,0Ch, 0Dh, 0Eh, 0Fh, 10h, 13h, 14h, 19h, 20h, 21h, 22h, 23h, 24h, 26h, 27h, 29h, 2Ch, 2Dh and C0h through FFh. All remaining operation codes for direct-access logical units are reserved for future standardization.

6.1.1 FORMAT UNIT command

The FORMAT UNIT command (see table 2) formats the medium into application client addressable logical blocks per the application client defined options. In addition, the medium may be certified and control structures may be created for the management of the medium and defects. There is no guarantee that the medium has or has not been altered.

Table 2 - FORMAT UNIT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (04h)							
1	Reserved				FmtData	CmpLst	Defect list format	
2	Vendor-specific							
3	(MSB)							
4	Interleave							(LSB)
5	Control							

The simplest mandatory form of the FORMAT UNIT command (with no format data) accomplishes medium formatting with little application client control overdefect management. The device server implementation determines the degree of defect management that is to be performed. Two additional mandatory forms of this command increase the application client's control over defect management. Several optional forms of this command further increase the application client's control over defect management, by allowing the application client to specify which defect list(s) are to be used, to specify defect locations (in several formats), to enable logical unit certification, and to specify what to do in the event that defect lists are not accessible.

The FORMAT UNIT command shall be rejected with RESERVATION CONFLICT status if the logical unit is reserved, or any extent reservation, from any application client, is active in the specified logical unit.

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

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

b) In response to the INQUIRY command, the device server shall respond as commanded.

c) In response to the REQUEST SENSE command, unless an error has occurred, the device server shall return a sense key of NOT READY and an additional sense code of LOGICAL UNIT NOT READY FORMAT IN PROGRESS, with the sense key specific bytes set for progress indication (as described in the SPC). Refer to the SPC for a description of deferred error handling that may occur during the format operation.

NOTE 1 It is recommended that MODE SELECT parameters (if any) be set prior to issuing the FORMAT UNIT command.

During the execution of the FORMAT UNIT command, the device server may perform a medium defect management algorithm (which can be controlled by the application client, using optional forms of this command). Four sources of defect location information (hereafter called defects) are defined as follows:

a) Primary defect list (Plist). This is the list of defects, usually supplied by the original manufacturer of the device or medium, that are considered permanent defects. The Plist is located outside of the application client-accessible logical block space. The Plist is accessible by the device server

(to reference while formatting), but it is not normally accessible by the application client except through the READ DEFECT DATA command. Once created, the original Plist shall not be subject to change.

b) Logical unit certification list (Clist). This list includes defects detected by the device server during an optional certification process executed during the FORMAT UNIT command. This list shall be added to the Glist.

c) Data defect list (Dlist). This list of defect descriptors may be supplied to the device server by the application client in the DATA OUT phase of the FORMAT UNIT command. This list shall be added to the Glist. The defect list length in the defect list header may be zero, in which case there is no Dlist.

d) Grown defect list (Glist). The Glist includes all defects sent by the application client or detected by the device server. The Glist does not include the Plist. If the CmpLst bit is zero, the Glist shall include Dlists provided to the device server during the previous and the current FORMAT UNIT commands. The Glist shall also include:
   a) defects detected by the format operation during medium certification;
   b) defects previously identified with a REASSIGN BLOCKS command;
   c) defects previously detected by the device server and automatically reallocated.

A format data (FmtData) bit of one indicates that the FORMAT UNIT parameter list (see table 3) shall be transferred during the DATA OUT phase. The DATA OUT phase consists of a defect list header (see table 4), followed by an initialization pattern descriptor, followed by zero or more defect descriptors. Each defect descriptor identifies a location on the medium that the device server shall map out of the user-accessible area.

Table 3 - FORMAT UNIT parameter list

Bit / Byte	7	6	5	4	3	2	1	0
	Defect list header							
	Initialization pattern descriptor (if any)							
	Defect descriptor(s) (if any)							
0	Defect descriptor 0							
n	(See specific table for length.)							
	.							
	.							
0	Defect descriptor x							
n	(See specific table for length.)							

The defect list header (see table 4) provides several optional format control bits. Device servers that implement these bits give the application client additional control over the use of the four defect sources, and the formatting operation. If the application client attempts to select any function not implemented by the device server, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

Table 4 - Defect list header

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	FOV	DPRY	DCRT	STPF	IP	DSP	Immed	VS
2	(MSB) Defect list length							
3								(LSB)

A FmtData bit of zero indicates that a DATA OUT phase shall not occur. The source of defect information is not specified.

A complete list (CmpLst) bit of one indicates that the defect list sent by the application client is a complete list of defects. Any existing defect list except the Plist shall be ignored by the device server. As a result, a new Glist is constructed that contains the Dlist (if it is sent by the application client), and the Clist (if certification is enabled). The device server may add any defects it detects during the format operation to this Dlist.

A CmpLst bit of zero indicates that the defect list sent by the application client is an addition to the existing list of defects. As a result a new Glist is constructed that contains the existing Glist, the Dlist (if it is sent by the application client), and the Clist (if certification is enabled). The device server may add any defects it detects during the format operation to this Dlist.

Table 5 defines the implementation requirements for the FORMAT UNIT command.

Table 5 - FORMAT UNIT defect descriptor format and requirements

FmtData	CmpLst	Defect list format	Defect list length	Type	Comments
0	0	000b	N/A	M	Vendor-specific
			BLOCK FORMAT		
1	0	000b	Zero	M	See notes (1) and (3)
1	1	000b	Zero	M	See notes (1) and (4)
1	0	000b	>0	O	See notes (2) and (3)
1	1	000b	>0	O	See notes (2) and (4)
			BYTES FROM INDEX FORMAT		
1	0	100b	Zero	O	See notes (1) and (3)
1	1	100b	Zero	O	See notes (1) and (4)
1	0	100b	>0	O	See notes (2) and (3)
1	1	100b	>0	O	See notes (2) and (4)
			PHYSICAL SECTOR FORMAT		
1	0	101b	Zero	O	See notes (1) and (3)
1	1	101b	Zero	O	See notes (1) and (4)
1	0	101b	>0	O	See notes (2) and (3)
1	1	101b	>0	O	See notes (2) and (4)
			VENDOR-SPECIFIC FORMAT		
1	0	110b			
1	1	110b			
All remaining codes are reserved.					
Key: M = Command implementation is mandatory. O = Command implementation is optional.					
NOTES 1 No Dlist is transferred to the device server during the DATA OUT phase. 2 A Dlist is transferred to the device server during the DATA OUT phase. Add the Dlist defects to the new Glist. 3 Use the existing Glist as a defect source. Add existing Glist defects to the new Glist. 4 Discard the existing Glist. Do not add existing Glist defects to the new Glist. 5 All the options described in this table cause a new Glist to be created during execution of the FORMAT UNIT command as described in the text.					

The defect list format field specifies which defect descriptor is used if the FmtData bit is one (see table 5).

The interleave field specifies the interleave that is used when performing the format operation. This allows the logical blocks to be related in a way that facilitates matching the transfer rate between the application client and the peripheral. An interleave of zero specifies that the device server use its default interleave. An interleave of one specifies that consecutive logical blocks be placed in contiguous ascending order. All other values are vendor-specific.

A format options valid (FOV) bit of zero indicates that the device server shall use its default settings for the DPRY, DCRT, STPF, IP and DSP bits (see below). The application client shall set these bits to zero. If any of these bits are not zero, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

A FOV bit of one indicates that the device server shall examine the setting of the DPRY, DCRT, STPF, IP and DSP bits. When the FOV bit is one, the DPRY, DCRT, STPF, IP and DSP bits are defined as follows.

A disable primary (DPRY) bit of zero indicates that the device server shall not use portions of the medium identified as defective in the primary defect Plist for application client addressable logical blocks. If the device server cannot locate the Plist or it cannot determine whether a Plist exists, it shall perform the action specified by the STPF bit. A DPRY bit of one indicates that the device server shall not use the Plist to identify defective areas of the medium. The Plist is not deleted.

A disable certification (DCRT) bit of zero indicates that the device server shall perform a vendor-specific medium certification operation to generate a Clist. A DCRT bit of one indicates that the device server shall not perform any vendor-specific medium certification process or format verification operation while executing the FORMAT UNIT command.

The stop format (STPF) bit controls the behaviour of the device server when one of the following events occurs:

a) The device server has been requested to use the primary defect list (DPRY is set to zero), or the grown defect list (CmpLst is set to zero) and the device server cannot locate the list nor determine whether the list exists.

b) The device server has been requested to use the primary defect list (DPRY is set to zero) or the grown defect list (CmpLst is set to zero), and the device server encounters an error while accessing the defect list.

A STPF bit of zero indicates that, if one or both of the above conditions occurs, the device server shall continue to execute the FORMAT UNIT command. The device server shall return CHECK CONDITION status at the completion of the FORMAT UNIT command. The sense key shall be set to RECOVERED ERROR and the additional sense code shall be set to either DEFECT LIST NOT FOUND if the first condition occurred, or DEFECT LIST ERROR if the second condition occurred.

A STPF bit of one indicates that, if one or both of the above conditions occurs, the device server shall terminate the FORMAT UNIT command with CHECK CONDITION status. The sense key shall be set to MEDIUM ERROR and the additional sense code shall be set to either DEFECT LIST NOT FOUND if the first condition occurred, or DEFECT LIST ERROR if the second condition occurred.

NOTE 2 The use of the FmtData bit, the CmpLst bit, and the defect header allow the application client to control the source of the defect lists used by the FORMAT UNIT command. Setting the defect list length to zero allows the application client to control the use of Plist and Clist without having to specify a Dlist.

An initialization pattern (IP) bit of one indicates that an initialization pattern descriptor (see 6.1.1.2) is included in the FORMAT UNIT parameter list immediately following the defect list header. An IP bit of zero indicates that an initialization pattern descriptor is not included and that the device server shall use its default initialization pattern.

A disable saving parameters (DSP) bit of one specifies that the device server shall not save the MODE SELECT savable parameters to non-volatile memory during the format operation. A DSP bit of zero specifies that the device server shall save all the MODE SELECT savable parameters for all application clients to non-volatile memory during the format operation. Pages that are not reported as savable are not affected by the DSP bit (i.e. if pages 03h & 04h are not returned with the PS bit set they may be saved even if DSP is cleared).

An immediate (Immed) bit of zero indicates that status shall be returned after the format operation has completed. An Immed bit value of one indicates that the device server shall return status as soon as the command descriptor block has been validated, and the entire defect list has been transferred.

The bit designated VS is vendor-specific.

The defect list length field in the defect list header specifies the total length in bytes of the defect descriptors that follow and does not include the initialization pattern descriptor or initialization pattern, if any. The length of the defect descriptors varies with the format of the defect list. The three formats for the defect descriptor(s) field in the defect lists are shown in 6.1.1.1.

6.1.1.1 Defect list formats

This clause describes the defect list formats used in the FORMAT UNIT, READ DEFECT DATA and translate page of the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands.

> **NOTE 3** The selected reporting format accounts for variables that impact the information in the returned data. For example, the specific location of a defect, while constant in angular and radial location on the logical unit, may change in reported location a format operation with different geometry parameters is performed. It is the responsibility of the application client to use a defect list format appropriate for the intended operation with the current or future geometry parameters. If the device server is able to detect that the selected defect list format would provide inconsistent results, the device server may return CHECK CONDITION status.

Each block format defect descriptor (see table 6) specifies a four-byte defective block address that contains the defect.

Table 6 - Defect descriptor - Block format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	Defective block address							
3								(LSB)

The defect list length is equal to four times the number of defect descriptors.

The defect descriptors should be in ascending order. More than one physical or logical block may be affected by each defect descriptor. A device server may return CHECK CONDITION if the defect descriptors are not in ascending order.

Each byte from index defect descriptor (see table 7 specifies the location of a defect that is no more than eight bytes long.

Table 7 - Defect descriptor - Bytes from index format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	Cylinder number of defect							
2								(LSB)
3	Head number of defect							
4	(MSB)							
	Defect bytes from index							
7								(LSB)

The defect list length is equal to eight times the number of defect descriptors.

Each descriptor is comprised of the cylinder number of defect, the head number of defect, and the defect bytes from index to the defect. The defect descriptors shall be in ascending order. The cylinder number of defect is the most significant part of the address and the defect bytes from index is the least significant part of the address. More than one physical or logical block may be affected by each defect. If the defect bytes from index has a value of FFFFFFFFh, this indicates that the entire track shall be considered defective.

Each physical sector defect descriptor (see table 8) specifies the location of a defect that is the length of a sector.

Table 8 - Defect descriptor - Physical sector format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				Cylinder number of defect				
2								(LSB)
3				Head number of defect				
4	(MSB)							
5				Defective sector number				
6								
7								(LSB)

The defect list length is equal to eight times the number of defect descriptors.

Each descriptor is comprised of a cylinder number of defect, the head number of defect, and the defective sector number. The defect descriptors shall be in ascending order. The cylinder number of defect is the most significant part of the address and the defective sector number is the least significant part of the address. More than one block may be affected by each defect descriptor. A defective sector number of FFFFFFFFh indicates that the entire track shall be considered defective.

6.1.1.2 Initialization pattern option

The initialization pattern option specifies that the logical blocks contain the specified initialization pattern. The initialization pattern descriptor (see table 9) is sent to the device server as part of the FORMAT UNIT parameter list.

Table 9 - Initialization pattern descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	IP modifier		Reserved					
1	Pattern type							
2	(MSB)							
3			Initialization pattern length					(LSB)
4								
n			Initialization pattern					

NOTE 4 The initialization pattern option is not intended for media analysis or certification. This option may only initialize the application client accessible area of the media to the specified pattern and may not write to any application client inaccessible areas of the disk.

The IP modifier field specifies the type and location of a header that modifies the initialization pattern (see table 10).

Table 10 - Initialization pattern modifier

IP modifier	Description
00b	No header. The device server shall not modify the initialization pattern.
01b	The device server shall overwrite the initialization pattern to write the logical block address in the first four bytes of the logical block. The logical block address shall be written with the most significant byte first.
10b	The device server shall overwrite the initialization pattern to write the logical block address in the first four bytes of each physical block contained within the logical block. The lowest numbered logical block or part thereof that occurs within the physical block is used. The logical block address shall be written with the most significant byte first.
11b	Reserved.

The initialization pattern type field (see table 11) indicates the type of pattern the device server shall use to initialize each logical block within the application client accessible portion of the medium. All bytes within a logical block shall be written with the initialization pattern. The initialization pattern is modified by the IP modifier field as described in table 10.

Table 11 - Initialization pattern type

Initialization pattern type	Description
00h	Use default pattern. (note 1)
01h	Repeat the initialization pattern as required to fill the logical block. (note 2)
02h - 7Fh	Reserved
80h - FFh	Vendor-specific

Notes
1) If the initialization pattern length is not zero 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 to INVALID FIELD IN PARAMETER LIST.
2) If the initialization pattern length is zero 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 to INVALID FIELD IN PARAMETER LIST.

The initialization pattern length field indicates the number of bytes contained in the initialization pattern. If the length exceeds the current logical block size the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST. The pattern is modified by the IP modifier field.

6.1.2 LOCK UNLOCK CACHE command

The LOCK UNLOCK CACHE command (see table 12) requests that the device server disallow or allow logical blocks within the specified range to be removed from the cache memory by the device server's cache replacement algorithm. Locked logical blocks may be written to the medium when modified, but a copy of the modified logical block shall remain in the cache memory.

Table 12 - LOCK UNLOCK CACHE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (36h)							
1	Reserved						Lock	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Number of blocks				(LSB)
9	Control							

A lock bit of one indicates that any logical block in the specified range that is currently present in the cache memory shall be locked into cache memory. Only logical blocks that are already present in the cache memory are actually locked. A lock bit of zero indicates that all logical blocks in the specified range that are currently locked into the cache memory shall be unlocked, but not necessarily removed.

A relative address (RelAdr) bit of one indicates that the logical block address field is a two's complement displacement. This negative or positive displacement shall be added to the logical block address last accessed on the logical unit to form the logical block address for this command. This feature is only available when linking commands. The feature requires that a previous command in the linked group have accessed a block of data on the logical unit.

A RelAdr bit of zero indicates that the logical block address field specifies the first logical block of the range of logical blocks to be operated on by this command.

The number of blocks specifies the total number of contiguous logical blocks within the range. A number of blocks field of zero indicates that all remaining logical blocks on the logical unit shall be within the range.

Multiple locks may be in effect from more than one application client. Locks from different application clients may overlap. An unlock of an overlapped area does not release the lock of another application client.

6.1.3 PRE-FETCH command

The PRE-FETCH command (see table 13) requests that the device server transfer the specified logical blocks to the cache memory. No data shall be transferred to the application client.

Table 13 - PRE-FETCH command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (34h)							
1	Reserved						Immed	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Transfer length				(LSB)
9	Control							

An immediate (Immed) bit of one indicates that status shall be returned as soon as the command descriptor block has been validated. An Immed bit of zero indicates that status shall be returned after the operation is complete.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The transfer length field specifies the number of contiguous logical blocks of data that shall be transferred to the logical unit's cache memory. A transfer length of zero indicates that the contiguous logical blocks up to and including the last logical block of the logical unit shall be transferred to the logical unit's cache memory. Any other value indicates the number of logical blocks that shall be transferred. The device server may elect to not transfer logical blocks that already are contained in the cache memory.

If the Immed bit is zero and the specified logical blocks were successfully transferred to the cache memory, the device server shall return CONDITION MET status. If the link bit (see 7.2.7) is one, the device server shall return INTERMEDIATE-CONDITION MET status.

If Immed is one, and the unlocked cache memory has sufficient capacity to accept all of the specified logical blocks, the device server shall return CONDITION MET status. If the link bit (see 7.2.7) is one, the device server shall return INTERMEDIATE-CONDITION MET status.

If Immed is one, and the unlocked cache memory does not have sufficient capacity to accept all of the specified logical blocks, the device server shall return GOOD status. The device server shall transfer to cache memory as many logical blocks as will fit. If the link bit (see 7.2.7) is one, the device server shall return INTERMEDIATE status.

6.1.4 PREVENT ALLOW MEDIUM REMOVAL command

The PREVENT ALLOW MEDIUM REMOVAL command (see table 14) requests that the device server enable or disable the removal of the medium in the logical unit. This mechanism is independent of logical unit reservations and the device server shall not allow medium removal if any application client currently has medium removal prevented.

Table 14 - PREVENT ALLOW MEDIUM REMOVAL command

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

The prevention of medium removal shall begin when any application client issues a PREVENT ALLOW MEDIUM REMOVAL command with a non-zero prevent value:

A Prevent value of 01b indicates that removal of medium from the data transport element shall be disabled.

A Prevent value of 10b indicates that removal of medium from a medium changer associated with the logical unit shall be disabled.

A Prevent of 11b indicates that removal of medium from the data transport element and from a medium changer associated with the logical unit shall be disabled.

Prevent values of 10b and 11b are only valid when the RMB bit and the Mchngr bit in the standard INQUIRY data are both equal to one.

The prevention of medium removal for the logical unit shall terminate:

a) after all application clients that have medium removal prevented issue PREVENT ALLOW MEDIUM REMOVAL commands with a prevent bit of zero,and the device server has successfully performed a synchronize cache operation;

b) upon the receipt of a BUS DEVICE RESET message from any application client; or

c) upon a hard RESET condition.

While a prevention of medium removal condition is in effect the logical unit shall inhibit mechanisms that normally allow removal of the medium by both an operator and by an attached medium changer.

6.1.5 READ(6) command

The READ(6) command (see table 15) requests that the device server transfer data to the application client. The most recent data value written in the addressed logical block shall be returned.

Table 15 - READ(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (08h)							
1	Reserved			(MSB)				
2	Logical block address							
3								(LSB)
4	Transfer length							
5	Control							

The cache control bits (see 6.1.6) are not provided for this command. Logical units with cache memory may have values for the cache control bits that affect the READ(6) command; however, no default value is defined by this International Standard. If explicit control is required, the READ(10) command should be used.

The logical block address field specifies the logical block at which the read operation shall begin.

The transfer length field specifies the number of contiguous logical blocks of data to be transferred. A transfer length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

6.1.6 READ(10) command

The READ(10) command (see table 16) requests that the device server transfer data to the application client. The most recent data value written in the addressed logical block shall be returned.

Table 16 - READ(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (28h)							
1	Reserved				DPO	FUA	Reserved	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Transfer length				(LSB)
9	Control							

A disable page out (DPO) bit of one indicates that the device server shall assign the logical blocks accessed by this command the lowest priority for being fetched into or retained by the cache. A DPO bit of one overrides any retention priority specified in the cache page (see 7.1.3.1). A DPO bit of zero indicates the priority shall be determined by the retention priority fields in the cache page. All other aspects of the algorithm implementing the cache memory replacement strategy are not defined by this International Standard.

> **NOTE 5** The DPO bit is used to control replacement of logical blocks in the cache memory when the host has information on the future usage of the logical blocks. If the DPO bit is set to one, the host knows the logical blocks accessed by the command are not likely to be accessed again in the near future and should not be put in the cache memory nor retained by the cache memory. If the DPO bit is zero, the host expects that logical blocks accessed by this command are likely to be accessed again in the near future.

A force unit access (FUA) bit of one indicates that the device server shall access the media in performing the command prior to returning GOOD status. Read commands shall access the specified logical blocks from the media (i.e. the data is not directly retrieved from the cache). In the case where the cache contains a more recent version of a logical block than the media, the logical block shall first be written to the media. Write commands shall not return GOOD status until the logical blocks have actually been written on the media (i.e. the data is not write cached).

An FUA bit of zero indicates that the device server may satisfy the command by accessing the cache memory. For read operations, any logical blocks that are contained in the cache memory may be transferred to the application client directly from the cache memory. For write operations, logical blocks may be transferred directly to the cache memory. GOOD status may be returned to the application client prior to writing the logical blocks to the medium. Any error that occurs after the GOOD status is returned is a deferred error, and information regarding the error is not reported until a subsequent command.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The transfer length field specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered an error. Any other value indicates the number of logical blocks that shall be transferred.

6.1.7 READ CAPACITY command

The READ CAPACITY command (see table 17) provides a means for the application client to request information regarding the capacity of the logical unit.

Table 17 - READ CAPACITY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (25h)							
1	Reserved							RelAdr
2	(MSB)							
3								
4	Logical block address							
5								(LSB)
6	Reserved							
7	Reserved							
8	Reserved							PMI
9	Control							

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The logical block address shall be zero if the PMI bit is zero. If the PMI bit is zero and the logical block address is not zero, the device server shall return a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN CDB.

A partial medium indicator (PMI) bit of zero indicates that the returned logical block address and the block length in bytes are those of the last logical block on the logical unit.

A PMI bit of one indicates that the returned logical block address and block length in bytes are those of the logical block address after which a substantial delay in data transfer will be encountered. This returned logical block address shall be greater than or equal to the logical block address specified by the RelAdr and logical block address fields in the command descriptor block.

> **NOTE 6** This function is intended to assist storage management software in determining whether there is sufficient space on the current track, cylinder, etc., to contain a frequently accessed data structure, such as a file directory or file index, without incurring an access delay.

The READ CAPACITY data (see table 18) shall be sent during the DATA IN phase of the command.

Table 18 - READ CAPACITY data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3	Returned logical block address							(LSB)
4	(MSB)							
7	Block length In bytes							(LSB)

6.1.8 READ DEFECT DATA command

The READ DEFECT DATA command (see table 19) requests that the device server transfer the medium defect data to the application client.

Table 19 - READ DEFECT DATA command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (37h)							
1	Reserved							
2	Reserved				Plist	Glist	Defect list format	
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Allocation length							
8								(LSB)
9	Control							

If the device server is unable to access any medium defect data, it shall terminate the command with CHECK CONDITION status. The sense key shall be set to either MEDIUM ERROR, if a medium error occurred, or NO SENSE, if the list does not exist; and the additional sense code shall be set to DEFECT LIST NOT FOUND.

> **NOTE 7** Some device servers may not be able to return medium defect data until after a FORMAT UNIT command has been completed successfully.

A primary defect list (Plist) bit of one request that the device server return the primary list of defects. A Plist bit of zero requests that the device server not return the primary list of defects.

A grown defect list (Glist) bit of one request that the device server return the grown defect list. A Glist bit of zero requests that the device server not return the grown defect list.

A Plist bit of one and a Glist bit of one requests that the device server return the primary and the grown defect lists. The order in which the lists are returned is vendor-specific. Whether the lists are merged or not is vendor-specific.

A Plist bit of zero and a Glist bit of zero requests that the device server return only the defect list header.

The defect list format field is used by the application client to indicate the preferred format for the defect list. This field is intended for those device servers capable of returning more than one format, as defined in the FORMAT UNIT command (see 6.1.1.2, defect list format). A device server unable to return the requested format shall return the defect list in its default format (see the defect list format field in the defect list header below).

If the requested defect list format and the returned defect list format are not the same, the device server shall transfer the defect data and then terminate the command with CHECK CONDITION status. The sense key shall be set to RECOVERED ERROR and the additional sense code shall be set to DEFECT LIST NOT FOUND.

The READ DEFECT DATA defect list (see table 20) contains a four-byte header, followed by zero or more defect descriptors.

Table 20 - READ DEFECT DATA defect list

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved				Plist	Glist	Defect list format	
2	(MSB)							
3	Defect list length							(LSB)
	Defect descriptor(s) (if any)							
0	Defect descriptor 0							
n	(See specific table for length.)							
	.							
	.							
0	Defect descriptor x							
n	(See specific table for length.)							

A Plist bit of one indicates that the data returned contains the primary defect list. A Plist bit of zero indicates that the data returned does not contain the primary defect list.

A Glist bit of one indicates that the data returned contains the grown defect list. A Glist bit of zero indicates that the data returned does not contain the grown defect list.

The defect list format field indicates the format of the defect descriptors returned by the device server. This field is defined in the FORMAT UNIT command (see 6.1.1.2).

> **NOTE 8** The use of the block format is not recommended. There is no universal model that sensibly defines the meaning of the logical block address of a defect. In the usual case, a defect that has been reassigned no longer has a logical block address.

Defect descriptors returned in the block format are vendor-specific. Defect descriptors returned in the physical sector format may or may not include defects in areas not accessible to the application client. Defect descriptors returned in bytes-from-index format shall comprise a complete list of defects. A complete list of defects may include defects in areas not within the capacity returned in the READ CAPACITY command.

The defect list length field specifies the length in bytes of the defect descriptors that follow. The defect list length is equal to four or eight times the number of defect descriptors, depending on the format of the returned descriptors (see 6.1.1.1).

If the allocation length is insufficient to transfer all of the defect descriptors, the defect list length shall not be adjusted to reflect the truncation. The device server shall not create CHECK CONDITION status. The application client is responsible for comparing the defect list length and the allocation length to ensure that a partial list was not received.

> **NOTE 9** The application client may determine the length of the defect list by sending the READ DEFECT DATA command with an allocation length of four. The device server will return the defect list header that contains the length of the defect list.

The defect descriptors may or may not be sent in ascending order. The application client may determine the exact number of defects by dividing the defect list length by the length of a single defect descriptor for the returned format.

6.1.9 READ LONG command

The READ LONG command (see table 21) requests that the device server transfer data to the application client. The data passed during the READ LONG command is vendor-specific, but shall include the data bytes and the ECC bytes recorded on the medium. The most recent data written in the addressed logical block shall be returned. READ LONG is independent of the READ-WRITE Error Recovery Page but does allow retries.

Table 21 - READ LONG command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (3Eh)							
1	Reserved						CORRCT	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Byte transfer length				(LSB)
9	Control							

NOTE 10 Any other bytes that can be corrected by ECC should be included (e.g. data synchronization mark within the area covered by ECC). It is not important for the ECC bytes to be at the end of the data bytes; however, they should be in the same order as they are on the media.

A corrected (CORRCT) bit of zero requests that a logical block to be read without any correction made by the device server. When the corrected (CORRCT) bit is 0, the command should result with GOOD status unless data cannot be transferred for some reason other than the data is non-correctable in which case the appropriate status and/or sense data shall be set. A CORRCT bit of one requests that the data be corrected by ECC before being transferred to the application client. If the DCR bit of the READ-WRITE Error Recovery Page is equal to one and the CORRCT bit equals one, the READ LONG command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The byte transfer length field should specify exactly the number of bytes of data that are available for transfer. If a non-zero byte transfer length does not exactly match the available data length, the device server shall terminate the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and an additional sense code set to INVALID FIELD IN CDB. The valid and ILI bits shall be set to one and the information field shall be set to the difference (residue) of the requested length minus the actual length in bytes. Negative values shall be indicated by two's complement notation.

A byte transfer length of zero indicates that no bytes shall be transferred and shall not be considered an error.

6.1.10 REASSIGN BLOCKS command

The REASSIGN BLOCKS command (see table 22) requests the device server to reassign the defective logical blocks to another area on the medium set aside for this purpose. The device server should also record the location of the defective logical blocks to the grown defect list if such a list is supported. More than one physical or logical block may be relocated by each defect descriptor sent by the application client. This command does not alter the contents or location of the Plist (see 6.1.1, FORMAT UNIT command).

Table 22 - REASSIGN BLOCKS command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (07h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Control							

The application client transfers a defect list that contains the logical block addresses to be reassigned. The device server shall reassign the physical medium used for each logical block address in the list. The data contained in the logical blocks specified in the defect list may be altered, but the data in all other logical blocks on the medium shall be preserved.

> **NOTE 11** The effect of specifying a logical block to be reassigned that previously has been reassigned is to reassign the block again. Over the life of the medium, a logical block can be assigned to multiple physical addresses until no more spare locations remain on the medium.

The REASSIGN BLOCKS defect list (see table 23) contains a four-byte header followed by one or more defect descriptors. The length of each defect descriptor is four bytes.

Table 23 - REASSIGN BLOCKS defect list

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	(MSB) Defect list length							
3								(LSB)
	Defect descriptor(s)							
0	(MSB) Defect logical block address							
3								(LSB)
	.							
0	(MSB) Defect logical block address							
3								(LSB)

The defect list length field specifies the total length in bytes of the defect descriptors that follow. The defect list length is equal to four times the number of defect descriptors and does not include the defect list header length.

The defect descriptor specifies a four-byte defect logical block address that contains the defect. The defect descriptors shall be in ascending order.

If the logical unit has insufficient capacity to reassign all of the logical blocks specified in the defect descriptors, the command shall terminate with CHECK CONDITION status, the sense key shall be set to HARDWARE ERROR and the additional sense code set to NO DEFECT SPARE LOCATION AVAILABLE.

If the logical unit is unable to successfully complete a REASSIGN BLOCKS command, the command shall terminate with CHECK CONDITION status with the appropriate sense information. The logical block address of the first defect descriptor not reassigned shall be returned in the command-specific information field of the sense data. If information about the first defect descriptor not reassigned is not available, or if all the defects have been reassigned, this field shall be set to FFFFFFFFh.

If the REASSIGN BLOCKS command failed due to an unexpected unrecoverable read error that would cause the loss of data in a block not specified in the defect list, the logical block address of the unrecoverable block shall be returned in the information field of the sense data and the valid bit shall be set to one.

NOTE 12 If the REASSIGN BLOCKS command returns CHECK CONDITION status and the sense data command-specific information field contains a valid logical block address, the application client should remove all defect descriptors from the defect list prior to the one returned in the command-specific information field. If the sense key is MEDIUM ERROR and the valid bit is one (the information field contains the valid block address) the application client should insert that new defective logical block address into the defect list and reissue the REASSIGN BLOCKS command with the new defect list. Otherwise, the application client should perform any corrective action indicated by the sense data and then reissue the REASSIGN BLOCKS command with the new defect list.

6.1.11 RELEASE command

The RELEASE command (see table 24) is used to release a previously reserved logical unit, or, if the extent release option is implemented, to release previously reserved extents within a logical unit.

Table 24 - RELEASE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (17h)							
1	Reserved				3rdPty	Third party device ID		Extent
2	Reservation identification							
3	Reserved							
4	Reserved							
5	Control							

The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-application client systems. A reservation may only be released by the application client that made it. It is not an error for an application client to attempt to release a reservation that is not currently valid, or is held by another application client. In this case, the device server shall return GOOD status without altering any other reservation.

> **NOTE 13** The reservation queuing option in X3.131-1986 (SCSI-1) has been removed as of SCSI-2.

6.1.11.1 Logical unit release

Implementation of logical unit release is mandatory. If the extent bit is zero, this command shall cause the device server to terminate all non-third- party logical unit and extent reservations that are active from the application client to the specified logical unit. The reservation ID field in the command descriptor block is ignored by the device server.

6.1.11.2 Extent release

Implementation of extent release is optional. If the extent bit is one and the extent release option is not implemented, then the RELEASE command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. This option shall be implemented if the extent reservation option (see 6.1.12.2) is implemented.

If the extent bit is one and the extent release option is implemented, this command shall cause any reservation from the requesting application client with a matching reservation identification to be terminated. Other reservations from the requesting application client shall remain in effect.

6.1.11.3 Third-party release

Implementation of third-party release is mandatory. Third-party release allows an application client to release a logical unit or extents within a logical unit that were previously reserved using third-party reservation (see 6.1.12.3). Third-party release shall be implemented and is intended for use in multiple-application client systems that use the COPY command.

If the third-party (3rdPty) bit is zero, then a third-party release is not requested. If the 3rdPty bit is one then the device server shall release the specified logical unit or extents, but only if the reservation was made using a third-party reservation by the application client that is requesting the release for the same SCSI logical unit as specified in the third-party device ID field.

If the 3rdPty bit is one the device server shall not modify the mode parameters for commands received from the third-party logical unit even if the device server implements the transfer of mode parameters with a third-party RESERVE command.

NOTE 14 If a device server implements independent storage of mode parameters for each application client, a third-party RESERVE command copies the current mode parameters for the application client that sent the RESERVE command to the current mode parameters for the application client specified as the third-party logical unit (usually a copy master logical unit). A unit attention condition notifies the third-party of the changed mode parameters due to the reservation. A successful third-party RELEASE command does not return the third-party logical units' current mode parameters back to their previous values. The third- party logical unit can issue MODE SENSE and MODE SELECT commands to query and modify the mode parameters.

6.1.12 RESERVE command

The RESERVE command (see table 25) is used to reserve a logical unit or, if the extent reservation option is implemented, extents within a logical unit.

Table 25 - RESERVE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (16h)							
1	Reserved			3rdPty	Third party device ID			Extent
2	Reservation identification							
3	(MSB) Extent list length							
4								(LSB)
5	Control							

The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-application client systems. The third-party reservation allows logical units or extents to be reserved for another specified SCSI logical unit.

NOTE 15 The reservation queuing option in X3.131-1986 (SCSI-1) has been removed as of SCSI-2.

6.1.12.1 Logical unit reservation

Implementation of logical unit reservation is mandatory. If the extent bit is zero, this command shall request that the entire logical unit be reserved for the exclusive use of the application client until the reservation is superseded by another valid RESERVE command from the application client that made the reservation or until released by a RELEASE command from the same application client that made the reservation, by a BUS DEVICE RESET message from any application client, by a hard RESET condition, or by a power on cycle. A logical unit reservation shall not be granted if the logical unit or any extent is reserved by another application client. It shall be permissible for an application client to reserve a logical unit that is currently reserved by that application client. If the extent bit is zero, the reservation identification and the extent list length shall be ignored.

If the logical unit, or any extent within the logical unit is reserved for another application client, the device server shall return RESERVATION CONFLICT status.

If, after honouring the reservation, any other application client attempts to perform any command on the reserved logical unit other than an INQUIRY, REQUEST SENSE, PREVENT ALLOW MEDIUM REMOVAL (with a prevent bit of zero), or a RELEASE command the command shall be rejected with RESERVATION CONFLICT status.

6.1.12.2 Extent reservation

Implementation of extent reservation is optional. The reservation identification field provides a means for an application client to identify each extent reservation. This allows an application client in a multiple tasking environment, to have multiple reservations outstanding. The reservation identification is used in the RELEASE command to specify which reservation is to be released. It is also used in superseding RESERVE commands to specify which reservation is to be superseded.

If the extent reservation option is implemented, then the extent release option (see 6.1.11.2) shall also be implemented. These options permit multiple extents within the logical unit to be reserved, each with a separate reservation type.

If the extent bit is one, and the extent reservation option is implemented, then the device server shall process the reservation request as follows:

a) The extent list shall be checked for the number of extents in the reservation request. If the extent list length is zero, no current reservations shall be changed, no new reservations shall be created, and this condition shall not be treated as an error. If the extent list contains more extents than are supported on the logical unit, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. If the extent list contains more extents than are currently available on the logical unit, then the device server shall return a RESERVATION CONFLICT status.

b) The extent list shall be checked for valid extent logical block addresses. If any logical block address is invalid for this logical unit, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST. The extent list shall be checked for invalid extent overlaps (as defined by reservation type) with other extent descriptors in the extent list and if invalid overlaps are found, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

c) If the requested reservation does not conflict with an existing reservation, the extents specified shall be reserved until superseded by another valid RESERVE command from the application client that made the reservation or until released by a RELEASE command from the same application client, by a BUS DEVICE RESET message from any application client, or by a hard RESET condition. If either of the last two conditions occur, the next command from each application client shall be terminated with CHECK CONDITION status and the sense key shall be set to UNIT ATTENTION.

d) If the reservation request conflicts with an existing reservation, then the device server shall return a RESERVATION CONFLICT status.

If the extent bit is one, and the extent reservation option is not implemented, then the RESERVE command shall be rejected with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

The size of the extent list shall be defined by the extent list length field. The extent list shall consist of zero or more descriptors as shown in table 26. Each extent descriptor defines an extent beginning at the specified logical block address for the specified number of blocks. If the number of blocks is zero, the extent shall begin at the specified logical block address and continue through the last logical block address on the logical unit.

Table 26 - Data format of extent descriptors

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved					RelAdr	Reservation type	
1	(MSB)							
2	Number of blocks							
3								(LSB)
4	(MSB)							
5	Logical block address							
6								
7								(LSB)

The reservation type field shall define the type of reservation in effect for the extent. The types of reservation are defined in table 27.

Table 27 - Reservation types

Reservation type	Description
00b	Read shared
01b	Write exclusive
10b	Read exclusive
11b	Exclusive access

a) Read exclusive. While this reservation is active, no other application client shall be permitted read operations to the indicated extent. This reservation shall not inhibit write operations from any application client or conflict with a write exclusive reservation; however, read exclusive, exclusive access, and read shared reservations that overlap this extent shall conflict with this reservation.

b) Write exclusive. While this reservation is active, no other application client shall be permitted write operations to the indicated extent. This reservation shall not inhibit read operations from any application client or conflict with a read exclusive reservation from any application client. This reservation shall conflict with write exclusive, exclusive access, and read shared reservations that overlap this extent.

c) Exclusive access. While this reservation is active, no other application client shall be permitted any access to the indicated extent. All reservation types that overlap this extent shall conflict with this reservation.

d) Read shared. While this reservation is active, no write operations shall be permitted by any application client to the indicated extent. This reservation shall not inhibit read operations from any application client or conflict with a read shared reservation. Read exclusive, write exclusive, and exclusive access reservations that overlap with this extent shall conflict with this reservation.

If the relative address bit is one, the logical block address in the extent descriptor shall be treated as a two's complement displacement. This displacement shall be added to the logical block address last accessed on the logical unit to form the logical block address for this extent. This feature is only available when linking commands and requires that a previous command in the linked group has accessed a logical block on the logical unit; if not, the RESERVE command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

If an application client attempts a command to a logical block that has been reserved and that access is prohibited by the reservation, the command shall not be performed and the command shall be terminated with a RESERVATION CONFLICT status. If a reservation conflict precludes any part of the command, none of the command shall be performed. COPY commands shall be terminated with CHECK CONDITION status and the sense key shall be set to DATA PROTECT if any part of the copy operation is prohibited by an active reservation. If any extent in a logical unit is reserved in any way, by any application client, a FORMAT UNIT command shall be rejected with a RESERVATION CONFLICT status.

6.1.12.3 Third-party reservation

Implementation of third-party reservation is mandatory. The third-party reservation for the RESERVE command allows an application client to reserve a logical unit or extents within a logical unit for another SCSI logical unit. This is intended for use in multiple-application client systems that use the COPY command. Third-party reservation is required.

If the third-party (3rdPty) bit is zero, then a third-party reservation is not requested. If the 3rdPty bit is one then the device server shall reserve the specified logical unit or extents for the SCSI logical unit specified in the third-party device ID field. The device server shall preserve the reservation until it is superseded by another valid RESERVE command from the application client that made the reservation or until it is released by the same application client, by a BUS DEVICE reset message from any application client, or a hard reset condition. The device server shall ignore any attempt to release the reservation made by any other application client.

If independent sets of parameters are implemented, a third party reservation shall cause the device server to transfer the set of parameters in effect for the application client of the RESERVE command to the parameters used for commands from the third party logical unit. Any subsequent command issued by the third-party logical unit is executed according to the mode parameters in effect for the application client that sent the RESERVE command.

> **NOTE 16** This transfer of the mode parameters is applicable to device server devices which store mode information independently for different application clients. This mechanism allows an application client to set the mode parameters of a device server for the use of a copy master (i.e. the third-party logical unit). The third-party copy master may subsequently issue a MODE SELECT command to modify the mode parameters.

6.1.12.4 Superseding reservations

Implementation of superseding reservations is mandatory. An application client that holds a current reservation (unit or extent) may modify that reservation by issuing another RESERVE command (unit or extent) to the same logical unit. The superseding RESERVE command shall release the previous reservation state (unit or extent) when the new reservation request is granted. If the superseding reservation is for an extent reservation and the current reservation is also an extent reservation, the current extent reservation identification value is used for the superseding reservation. The current reservation shall not be modified if the superseding reservation request cannot be granted. If the superseding reservation cannot be granted because of conflicts with a previous reservation (other than the reservation being superseded), then the device server shall return RESERVATION CONFLICT status.

> **NOTE 17** Superseding reservations allow the SCSI logical unit ID to be changed on a reservation using the third-party reservation option. This capability is necessary for certain situations when using COMPARE, COPY, and COPY AND VERIFY commands.

6.1.13 REZERO UNIT command

The REZERO UNIT command (see table 28) requests that the device server set the logical unit to a specific state. See vendor specifications for details.

Table 28 - REZERO UNIT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (01h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Control							

6.1.14 SEARCH DATA commands

The SEARCH DATA commands (see table 29) search one or more logical blocks for equality or inequality to a data pattern. The concept of records within a logical block is used to allow multiple records within a logical block to be searched.

Table 29 - SEARCH DATA commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (31h 30h 32h)							
1	Reserved			Invert	Reserved		SpnDat	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Number of blocks to search				(LSB)
9	Control							

The invert bit determines whether the search condition is to be inverted. See 6.1.14.1 through 6.1.14.3 for a description of the search conditions for the individual SEARCH DATA commands.

A spanned data (SpnDat) bit of zero indicates that each record shall be wholly contained within a single block. Any space at the end of a block that is smaller than the record length is ignored by the SEARCH DATA commands. A SpnDat bit of one indicates that records span block boundaries (i.e. record may start in one block and end in the next or a subsequent block.

The number of blocks to search field specifies the maximum number of contiguous logical blocks to be searched. A value of zero indicates that no logical blocks shall be searched. This condition shall not be considered an error. Any other value indicates the maximum number of logical blocks that shall be searched.

A link bit (see 7.2.7) of zero indicates a non-linked command and if the search is satisfied, the command shall be terminated with a CONDITION MET status. A REQUEST SENSE command can then be issued to determine the logical block address and record offset of the matching record. If the search is not satisfied and no error occurs, the command shall be terminated with GOOD status.

A link bit (see 7.2.7) of one indicates a command is linked to the SEARCH DATA command and if the search is satisfied, INTERMEDIATE-CONDITION MET status is returned and the next command is executed. If the RelAdr bit in the next command is one, the logical block address of the next command is used as a displacement from the logical block address at which the search was satisfied. If a linked search is not satisfied, the command is terminated with CHECK CONDITION status. A REQUEST SENSE command may then be issued.

A REQUEST SENSE command following a satisfied SEARCH DATA command shall:

a) return a sense key of EQUAL if the search was satisfied by an exact match. If the search was satisfied by an inequality then a sense key of NO SENSE shall be returned.

b) return the valid bit set to one.

c) return the logical block address of the logical block containing the first matching record in the information field.

d) return the record offset of the matching record in the command- specific information field.

A REQUEST SENSE command following a SEARCH DATA command that is not satisfied shall:

a) return a sense key of NO SENSE, if no errors occurred during the command execution.

b) return the valid bit set to zero.

The SEARCH DATA parameter list (see table 30) contains a fourteen-byte header, followed by one or more search argument descriptors.

Table 30 - SEARCH DATA parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3	Logical record length							(LSB)
4	(MSB)							
7	First record offset							(LSB)
8	(MSB)							
11	Number of records							(LSB)
12	(MSB)							
13	Search argument length							(LSB)
	Search argument descriptor(s)							
0	(MSB)							
3	Displacement							(LSB)
4	(MSB)							
5	Pattern length							(LSB)
6								
n	Pattern							

The logical record length field specifies the record length in bytes.

The first record offset field specifies the number of bytes that shall be ignored in the first logical block before the search begins. If the value of the first record offset field shall is larger than the logical block length the device server shall terminate the command with a CHECK CONDITION status, set the sense key to ILLEGAL REQUEST and set the additional sense code to INVALID FIELD IN PARAMETERS LIST. Subsequent logical blocks shall be searched beginning with the first byte in the logical block. This permits one or more records to be skipped initially.

The number of records field specifies the maximum number of records that shall be searched by this command. An search shall terminate when the search pattern is found or when the number of records is exhausted or when the number of blocks to search is exhausted.

The search argument length field specifies the length in bytes of all the search argument descriptors that follow.

NOTE 18 Since the pattern length can vary, there is no fixed multiple of the search argument descriptor to determine the search argument length.

The search argument descriptors specify one or more search conditions to execute within a single record in order to satisfy the search. Each search argument descriptor is made up of a displacement field, a pattern length field, and a pattern field.

The displacement field specifies the displacement in bytes of the first byte of the data to be compared from the start of the logical record.

The pattern length field specifies the length in bytes of the pattern that follows.

The pattern field specifies the data to compare to the logical record.

6.1.14.1 SEARCH DATA EQUAL command

The SEARCH DATA EQUAL command (see table 29, operation code 31h) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being equal to the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being not equal to the data in the pattern. (See 6.1.14.)

6.1.14.2 SEARCH DATA HIGH command

The SEARCH DATA HIGH command (see table 29, operation code 30h) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being greater than the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being less than or equal to the data in the pattern. (See 6.1.14.)

6.1.14.3 SEARCH DATA LOW command

The SEARCH DATA LOW command (see table 29, operation code 32h) shall be satisfied by the first logical record searched that contains data that satisfies all of the search argument descriptor(s). If the invert bit in the command descriptor block is zero, the search argument descriptor(s) shall be satisfied by data in the logical record being less than the data in the pattern. If the invert bit is one, the search argument descriptor(s) shall be satisfied by data in the logical record being greater than or equal to the data in the pattern. (See 6.1.14.)

6.1.15 SEEK(6) and SEEK(10) commands

The SEEK(6) (see table 31) and SEEK(10) (see table 32) commands request that the logical unit seek to the specified logical block address.

Table 31 - SEEK(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (0Bh)							
1	Reserved			(MSB)				
2	Logical block address							
3								(LSB)
4	Reserved							
5	Control							

Table 32 - SEEK(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Bh)							
1	Reserved							
2	(MSB)							
3	Logical block address							
4								
5								(LSB)
6	Reserved							
7	Reserved							
8	Reserved							
9	Control							

6.1.16 SET LIMITS command

The SET LIMITS command (see table 33) defines the range within which subsequent linked commands may operate. A second SET LIMITS command may not be linked to a chain of commands in which a SET LIMITS command has already been issued.

Table 33 - SET LIMITS command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (33h)							
1	Reserved						RdInh	WrInh
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Number of blocks				(LSB)
9	Control							

The read inhibit (RdInh) bit of one indicates that read operations within the range shall be inhibited.

A write inhibit (WrInh) bit of one indicates that write operations within the range shall be inhibited.

The logical block address field specifies the starting address for the range.

The number of blocks field specifies the number of logical blocks within the range. A number of blocks of zero indicates that the range shall extend to the last logical block on the logical unit.

Any attempt to access outside of the restricted range or any attempt to perform an inhibited operation within the restricted range shall not be performed. The command shall be terminated with CHECK CONDITION status and the sense key shall be set to DATA PROTECT. A second SET LIMITS command within a linked list of commands shall be rejected with CHECK CONDITION status and the sense key shall be set to DATA PROTECT.

6.1.17 START STOP UNIT command

The START STOP UNIT command (see table 34) requests that the device server enable or disable the logical unit for media access operations.

Table 34 - START STOP UNIT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (1Bh)							
1	Reserved							Immed
2	Reserved							
3	Reserved							
4	Power Conditions				Reserved		LoEj	Start
5	Control							

An immediate (Immed) bit of one indicates that status shall be returned as soon as the command descriptor block has been validated. An Immed bit of zero indicates that status shall be returned after the operation is completed.

The power conditions field requests the logical unit to be placed into the power condition defined in table 35. If any bit is set in this field then the Start and the LoEj bits shall be ignored.

There shall be no indication from the logical unit to any device that it has entered the requested power condition.

If the START STOP UNIT command is issued with the Power Conditions field set to 1, 2, 3, or 5 the logical unit shall:

a) change power conditions only on receipt of another START STOP UNIT command or a hard reset.

b) suspend any Power Condition timers (see table 34) that are active on receipt of the START STOP UNIT command until another START STOP UNIT command is received which returns control of the power condition to the logical unit or a hard reset occurs.

c) terminate any command received which requires more power than allowed by the START STOP UNIT commands most resent power condition setting with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to LOW POWER CONDITION ACTIVE.

The START STOP UNIT command is issued with the Power Conditions field set Ah or Bh the logical unit shall:

a) force the selected timer(s) to zero. Forcing the timer(s) to zero shall place the logical unit into the same power condition that would have occurred if the timer(s) would have normally timed out. After the timer(s) are set to zero control the power conditions is returned to the logical unit.

b) terminate any START STOP UNIT command which selects a timer which is not supported by the logical unit or a timer which has been disabled with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

It is not an error to request the logical unit be placed into a power condition which already exists.

Table 35 - Power Conditions

Code	Description
0h	No change in power conditions or in which device is controlling power conditions
1h	Place device into the Active condition
2h	Place device into Idle condition
3h	Place device into Standby condition
4h	Reserved
5h	Place device into Sleep condition
6h	Reserved
7h	Give control of power conditions to logical unit
8h–9h	Reserved
Ah	Force Idle Condition Timer to zero
Bh	Force Standby Condition Timer to zero
Ch–Fh	Reserved

In the Sleep condition the device server shall only respond to a bus reset condition. When a target has multiple logical units attached it shall enter the Sleep condition only after all the logical units have been placed into a Sleep condition.

A load eject (LoEj) bit of zero requests that no action be taken regarding loading or ejecting the medium. A LoEj bit of one requests that the medium shall be unloaded if the start bit is zero. A LoEj bit of one requests that the medium is to be loaded if the start bit is one.

A start bit of one requests the logical unit be made ready for use. A start bit of zero requests that the logical unit be stopped (media cannot be accessed by the application client).

Logical units that contain cache memory shall implicitly perform a SYNCHRONIZE CACHE command for the entire medium prior to executing the STOP UNIT command.

6.1.18 SYNCHRONIZE CACHE command

The SYNCHRONIZE CACHE command (see table 36) ensures that logical blocks in the cache memory, within the specified range, have their most recent data value recorded on the physical medium. If a more recent data value for a logical block within the specified range exists in the cache memory than on the physical medium, then the logical block from the cache memory shall be written to the physical medium. Logical blocks are not necessarily removed from the cache memory as a result of the synchronize cache operation.

Table 36 - SYNCHRONIZE CACHE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (35h)							
1	Reserved						Immed	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Number of blocks				(LSB)
9	Control							

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 zero indicates that the status shall not be returned until the operation has been completed. If the Immed bit is one and the device server does not support it then the command shall terminate with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The number of blocks field specifies the total number of contiguous logical blocks within the range. A number of blocks of zero indicates that all remaining logical blocks on the logical unit shall be within the range.

A logical block within the specified range that is not in cache memory is not considered an error.

6.1.19 VERIFY command

The VERIFY command (see table 37) requests that the device server verify the data written on the medium.

Table 37 - VERIFY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Fh)							
1	Reserved			DPO	Reserved	Reserved	BytChk	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Verification length				(LSB)
9	Control							

If the MODE SELECT command is implemented, and the verify error recovery parameters page is also implemented, then the current settings in that page specifies the verification error criteria. If the verify error recovery parameters page is not implemented, then the verification criteria is vendor-specific.

A byte check (BytChk) bit of zero causes a medium verification to be performed with no data comparison. A BytChk bit of one causes a byte-by- byte compare of data written on the medium and the data transferred from the application client. If the compare is unsuccessful for any reason, the device server shall return CHECK CONDITION status with the sense key set to MISCOMPARE.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The verification length field specifies the number of contiguous logical blocks of data that shall be verified. A transfer length of zero indicates that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be verified.

See 6.1.6 for a description of the cache control bit (DPO).

6.1.20 WRITE(6) command

The WRITE(6) command (see table 38) requests that the device server write the data transferred by the application client to the medium.

Table 38 - WRITE(6) command

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

The cache control bits (see 6.1.6) are not provided for this command. Logical units with cache memory may have values for the cache control bits which may affect the WRITE(6) command, however no default value is defined by this International Standard. If explicit control is required, the WRITE(10) command should be used.

The logical block address field specifies the logical block at which the write operation shall begin.

The transfer length field specifies the number of contiguous logical blocks of data to transferred. A transfer length of zero indicates that 256 logical blocks shall be transferred. Any other value indicates the number of logical blocks that shall be transferred.

6.1.21 WRITE(10) command

The WRITE(10) command (see table 39) requests that the device server write the data transferred by the application client to the medium.

Table 39 - WRITE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Ah)							
1	Reserved			DPO	FUA	Reserved	Reserved	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Transfer length				(LSB)
9	Control							

See READ(10) command (6.1.6) for a definition of the cache control bits (DPO and FUA).

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The transfer length field specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

6.1.22 WRITE AND VERIFY command

The WRITE AND VERIFY command (see table 40) requests that the device server write the data transferred from the application client to the medium and then verify that the data is correctly written. The data is only transferred once from the application client to the device server.

Table 40 - WRITE AND VERIFY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Eh)							
1	Reserved			DPO	Reserved	Reserved	BytChk	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Transfer length				(LSB)
9	Control							

If the MODE SELECT command is implemented, and the verify error recovery page is also implemented (see 7.1.3.8), then the current settings in that page along with the AWRE bit from the read-write error recovery page specify the verification error criteria. If these pages are not implemented, then the verification criteria is vendor-specific.

A byte check (BytChk) bit of zero requests a medium verification to be performed with no data comparison. A BytChk bit of one requests a byte- by-byte compare of data written on the medium and the data transferred from the application client. If the compare is unsuccessful for any reason, the device server shall return CHECK CONDITION status with the sense key set to MISCOMPARE.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

See WRITE(10) command (see 6.1.21) for a definition of the transfer length field.

See 6.1.6 for a description of the cache control bit (DPO).

> **NOTE 19** The WRITE AND VERIFY command specifically states that the data are not to be transferred twice (i.e. once for the write pass, and once for the verify pass) when performing a byte compare. If there is a need for two transfers to occur (e.g. to ensure the integrity of the path to the media), then the application client should issue a WRITE command with a LINK bit of one followed by a VERIFY command with a BytCmp bit of one, transferring the same data on each command.

6.1.23 WRITE LONG command

The WRITE LONG command (see table 41) requests that the device server write the data transferred by the application client to the medium. The data passed during the WRITE LONG command is implementation specific, but shall include the data bytes and the ECC bytes.

Table 41 - WRITE LONG command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (3Fh)							
1	Reserved							RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Byte transfer length				(LSB)
9	Control							

NOTE 20 Any other bytes that can be corrected by ECC should be included (e.g. a data synchronization mark within the area covered by ECC). The READ LONG command is usually issued before issuing a WRITE LONG command. The WRITE LONG data should be the same length and in the same order as the data returned by the READ LONG command.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The byte transfer length field should specify the number of bytes of data that the device server would return for the READ LONG command. If a non-zero byte transfer length does not exactly match the data length the device server would return for the READ LONG command, then the device server shall terminate the command with CHECK CONDITION status and a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB. The ILI and valid bits shall be set to one and the information field shall be set to the difference (residue) of the requested length minus the actual length in bytes. Negative values shall be indicated by two's complement notation. A transfer length of zero indicates that no bytes shall be transferred and shall not be considered an error.

6.1.24 WRITE SAME command

The WRITE SAME command (see table 42) requests that the device server write the single block of data transferred by the application client to the medium multiple times.

Table 42 - WRITE SAME command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (41h)							
1	Reserved					PBdata	LBdata	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Number of blocks				(LSB)
9	Control							

NOTE 21 This command is useful if large areas of the medium need to be written, prepared for certification, or otherwise initialized without the application client having to transfer all the data.

A logical block data (LBdata) bit of one requests that the device server replace the first four bytes of the data to be written to the current logical block with the logical block address of the block currently being written.

A physical block data (PBdata) bit of one requests that the device server replace the first eight bytes of the data to be written to the current physical sector with the physical address of the sector currently being written using the physical sector format (see 6.1.1.1).

If PBdata and LBdata are one the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST.

See 6.1.2 for a definition of the RelAdr bit and the logical block address field.

The number of blocks field specifies the number of contiguous logical blocks to be written. A number of blocks field of zero requests that all the remaining logical blocks on the medium be written.

6.2 Commands for optical memory logical units

The commands for optical memory logical units shall be as shown in table 43.

Table 43- Commands for optical memory logical units

Command name	Operation code	Type	Subclause
CHANGE DEFINITION	40h	O	SPC
COMPARE	39h	O	SPC
COPY	18h	O	SPC
COPY AND VERIFY	3Ah	O	SPC
ERASE (10)	2Ch	O	6.2.1
ERASE (12)	ACh	O	6.2.2
FORMAT UNIT	04h	O	6.1.1
INQUIRY	12h	M	SPC
LOCK-UNLOCK CACHE	36h	O	6.1.2
LOG SELECT	4Ch	O	SPC
LOG SENSE	4Dh	O	SPC
MEDIUM SCAN	38h	O	6.2.3
MODE SELECT(6)	15h	O	SPC
MODE SELECT(10)	55h	O	SPC
MODE SENSE(6)	1Ah	O	SPC
MODE SENSE(10)	5Ah	O	SPC
MOVE MEDIUM	A5h	O	SMC
PORT STATUS	11h	M[1]	SPC
PRE-FETCH	34h	O	6.1.3
PREVENT-ALLOW MEDIUM REMOVAL	1Eh	O	6.1.4
READ(6)	08h	O	6.1.5
READ(10)	28h	M	6.1.6
READ(12)	A8h	O	6.2.4
READ BUFFER	3Ch	O	SPC
READ CAPACITY	25h	M	6.1.7
READ DEFECT DATA (10)	37h	O	6.1.8
READ DEFECT DATA (12)	B7h	O	6.2.5
READ ELEMENT STATUS	B8h	O	SMC
READ GENERATION	29h	O	6.2.6
READ LONG	3Eh	O	6.1.9
READ UPDATED BLOCK	2Dh	O	6.2.7
REASSIGN BLOCKS	07h	O	6.1.10
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	SPC
RELEASE	17h	M	6.1.11
REQUEST SENSE	03h	M	SPC
RESERVE	16h	M	6.1.12
REZERO UNIT	01h	O	6.1.13
SEARCH DATA EQUAL (10)	31h	O	6.1.14
SEARCH DATA EQUAL (12)	B1h	O	6.2.8
SEARCH DATA HIGH (10)	30h	O	6.1.14
SEARCH DATA HIGH (12)	B0h	O	6.2.8
SEARCH DATA LOW (10)	32h	O	6.1.14
SEARCH DATA LOW (12)	B2h	O	6.2.8
SEEK(6)	0Bh	O	6.1.15
SEEK(10)	2Bh	O	6.1.15
SEND DIAGNOSTIC	1Dh	M	SPC
SET LIMITS (10)	33h	O	6.1.16
SET LIMITS (12)	B3h	O	6.2.9
START STOP UNIT	1Bh	O	6.1.17
SYNCHRONIZE CACHE	35h	O	6.1.18
TEST UNIT READY	00h	M	SPC
UPDATE BLOCK	3Dh	O	6.2.10
VERIFY (10)	2Fh	O	6.2.11
VERIFY (12)	AFh	O	6.2.12
WRITE(6)	0Ah	O	6.1.20
WRITE(10)	2Ah	M	6.2.13

WRITE(12)	AAh	O	6.2.14
WRITE AND VERIFY (10)	2Eh	O	6.2.15
WRITE AND VERIFY (12)	AEh	O	6.2.16
WRITE BUFFER	3Bh	O	SPC
WRITE LONG	3Fh	O	6.1.23
Key: M = Command implementation is mandatory.			
O = Command implementation is optional.			
SPC = SCSI-3 Primary Commands			
SMC = SCSI-3 Medium Changer Command Set			
Notes: [1] Mandatory only if the dual port option is implemented.			

The following codes are vendor-specific: 20h, 21h, 22h, 23h, and C0h through FFh. All remaining codes for optical memory logical units are reserved for future standardization.

6.2.1 ERASE(10) command

The ERASE(10) command (see table 44) requests that the device server erase the specified number of blocks starting at the specified logical block address on the medium. As used here, erased means either the medium shall be erased, or a pattern shall be written on the medium that appears to the device server as no data present. The blocks erased shall be considered blank for purposes of blank checking (see 5.2). The previous data recorded on the medium, if any, shall not be recoverable.

Table 44 - ERASE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Ch)							
1	Reserved					ERA	Reserved	RelAdr
2	(MSB)							
3								
4	Logical block address							
5								(LSB)
6	Reserved							
7	(MSB)							
8	Transfer length							(LSB)
9	Control							

The erase all (ERA) bit set to one indicates that all remaining blocks on the medium shall be erased. If the ERA bit is set to one and if the number of blocks is not zero, the device server shall return CHECK CONDITION, and the sense key shall be set to ILLEGAL REQUEST, with an additional sense code of INVALID FIELD IN CDB.

See 6.1.2 for a description of the RelAdr bit and logical block address field.

The transfer length specifies the number of contiguous logical blocks that shall be erased when the ERA bit is zero. If the ERA bit is zero a transfer length of zero indicates that no blocks shall be erased. This condition shall not be considered an error and no data shall be erased. Any other value indicates the number of logical blocks that shall be erased.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict (see 6.1.12) exists and no data shall be erased.

> **NOTE 22** This command allows the user to separate the erase and write operations. This may increase system performance in certain applications.

6.2.2 ERASE(12) command

The ERASE(12) command (see table 45) requests that the device server erase the specified number of blocks starting at the specified logical block address on the medium.

Table 45 - ERASE(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (ACh)							
1	Reserved					ERA	Reserved	RelAdr
2	(MSB)							
3								
4	Logical block address							
5								(LSB)
6	(MSB)							
7								
8	Transfer length							
9								(LSB)
10	Reserved							
11	Control							

See the ERASE(10) command (6.2.1) for a description of the fields in this command.

6.2.3 MEDIUM SCAN command

The MEDIUM SCAN command (see table 46) requests that the device server to scan the medium for a contiguous set of written or blank logical blocks.

Table 46 - MEDIUM SCAN command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (38h)							
1	Reserved			WBS	ASA	RSD	PRA	RelAdr
2	(MSB) Logical block address							
3								
4								
5								(LSB)
6	Reserved							
7	Reserved							
8	Parameter list length							
9	Control							

A written block search (WBS) bit of zero indicates that the scan is for blank blocks. A WBS bit of one indicates that the scan is for written blocks.

An advanced scan algorithm (ASA) bit of zero indicates that the scan area is scanned in sequential order (as selected by the RSD bit). An ASA bit of one indicates to the device server that the written and blank areas within the scan area form contiguous extents (as opposed to scattered blocks). This indication is advisory to the device server.

> **NOTE 23** The purpose of this bit is to allow the device server to use a more advanced algorithm (such as a binary search) to locate the requested blocks.

A reverse scan direction (RSD) bit of zero indicates the scan shall begin with the first logical block of the scan area. A RSD bit of one indicates the scan shall begin with the last logical block of the scan area.

A partial results acceptable (PRA) bit of zero indicates that the scan shall not be considered satisfied until a contiguous set of blocks is found within the scan area that is at least equal in size to the number of blocks requested, and meets the other criteria specified in the command descriptor block. A PRA bit of one indicates that the scan may be satisfied by a contiguous set of blocks within the scan area that is less than the number of blocks requested, and meets the other criteria specified in the command descriptor block.

See 6.1.2 for a description of the RelAdr bit and logical block address field.

The parameter list length specifies the length in bytes of the parameter list that shall be transferred during the DATA OUT phase. A parameter list length of zero indicates that the number of blocks requested field has a value of one, and the number of blocks to scan field has a value of zero. This condition shall not be considered an error. The contents of the parameter list are specified in table 47.

Table 47 - MEDIUM SCAN parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				Number of blocks requested				
2								
3								(LSB)
4	(MSB)							
5				Number of blocks to scan				
6								
7								(LSB)

A link bit of zero indicates a non-linked command; if the scan is satisfied, the command shall be terminated with a CONDITION MET status. A REQUEST SENSE command can then be issued to determine the starting logical block address of the area that meets the request. If the scan is not satisfied and no error occurs, the command shall be terminated with GOOD status.

A link bit of one indicates that a command is linked to the MEDIUM SCAN command; if the search is satisfied, CONDITION MET status is returned and the next command is executed. If the RelAdr bit in the next command is one, the logical block address of the next command is used as a displacement from the logical block address at which the search was satisfied. If a linked scan is not satisfied, the command is terminated with a CHECK CONDITION status. A REQUEST SENSE command may then be issued.

A REQUEST SENSE command following a satisfied MEDIUM SCAN command shall:

a) return a sense key of EQUAL if the scan was satisfied by a contiguous set of blocks equal in size to the number of blocks requested. If the PRA bis is one and the scan was satisfied by a contiguous set of blocks less than the number of blocks requested, then a sense key of NO SENSE shall be returned;

b) return the valid bit set to one;

c) return the logical block address of the first logical block of the contiguous set of blocks that satisfied the scan criteria in the information bytes;

d) return the number of contiguous logical blocks meeting the scan criteria in the command specific information bytes.

A REQUEST SENSE command following an unsatisfied MEDIUM SCAN command shall:

a) return a sense key of NO SENSE if no errors occurred during the command execution;

b) return the valid bit set to zero.

The number of blocks requested field specifies the number of blocks that meet the specified requirements. The number of blocks requested field, if set to zero, indicates that the scan shall not take place. This shall not be considered an error condition.

The number of blocks to scan field specifies the length in blocks of the area to be scanned on the medium. The number of blocks to scan field, if set to zero, indicates that the scan shall continue for all remaining blocks on the medium or until the scan is satisfied. See 5.2.2 for a description of error reporting.

6.2.4 READ(12) command

The READ(12) command (see table 48) requests that the device server transfer data to the application client from the medium. See the READ(10) command (6.1.6) for a complete description of the fields in this command.

Table 48 - READ(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (A8h)							
1	Reserved			DPO	FUA	Reserved		RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	(MSB)							
7								
8				Transfer length				
9								(LSB)
10	Reserved							
11	Control							

6.2.5 READ DEFECT DATA(12) command

The READ DEFECT DATA(12) command (see table 49) requests that the device server transfer the medium defect data to the application client. See the READ DEFECT DATA(10) command (6.1.8) for a description of the fields in this command.

Table 49 - READ DEFECT DATA(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (B7h)							
1	Reserved			Plist	GList	Defect list format		
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	(MSB)							
7				Allocation length				
8								
9								(LSB)
10	Reserved							
11	Control							

The READ DEFECT DATA(12) list header (see table 50) contains an eight byte header, followed by zero or more defect descriptors.

Table 50 - READ DEFECT DATA(12) list header

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved			PList	GList	Defect list format		
2	Reserved							
3	Reserved							
4	(MSB)							
5				Defect list length				
6								
7								(LSB)
Defect descriptors								
0								
n								

See the description of the READ DEFECT DATA defect list (see 6.1.1.1) for a description of the fields in this header.

6.2.6 READ GENERATION command

The READ GENERATION command (see table 51) requests that the device server transfer to the application client the maximum generation address for the logical block specified.

Table 51 - READ GENERATION command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (29h)							
1	Reserved							RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	Reserved							
8	Allocation length							
9	Control							

See 6.1.2 for a description of the RelAdr bit and logical block address field.

The READ GENERATION data is defined in table 52.

Table 52 - Maximum generation data block

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				Maximum generation address				(LSB)
2	Reserved							
3	Reserved							

The maximum generation address field defines the maximum generation address available for the logical block address specified.

6.2.7 READ UPDATED BLOCK(10) command

The READ UPDATED BLOCK(10) command (see table 53) requests that the device server transfer data to the application client from the specified generation and logical block.

Table 53 - READ UPDATED BLOCK(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Dh)							
1	Reserved				DPO	FUA	Reserved	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Latest	(MSB)						
7				Generation address				(LSB)
8	Reserved							
9	Control							

See 6.1.6 for a description of the cache control bits (DPO and FUA).
See 6.1.2 for a description of the RelAdr bit and logical block address field.

One block of data is transferred during the DATA IN phase.

The latest bit determines the meaning of the generation address field. A latest bit of zero indicates that the generation address is specified relative to the first generation of the block; generation address zero specifies the first generation. Increasing generation addresses specify later generations.

A latest bit of one indicates that the generation address is specified relative to the latest generation of the block; generation address zero specifies the most recent generation. Increasing generation addresses specify earlier generations.

If the requested generation does not exist, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to BLANK CHECK, and the additional sense code shall be set to GENERATION DOES NOT EXIST.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict (see 6.1.12) exists and no data shall be transferred.

6.2.8 SEARCH DATA(12) commands

The SEARCH DATA(12) commands (see table 54) search one or more logical blocks for equality or inequality to a data pattern. See the SEARCH DATA(10) commands (6.1.14) for a description of the fields in this command.

Table 54 - SEARCH DATA(12) commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (B0h, B1h, B2h)							
1	Reserved			Invert	Reserved		SpnDat	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	(MSB)							
7								
8				Transfer length				
9								(LSB)
10	Reserved							
11	Control							

6.2.9 SET LIMITS(12) command

The SET LIMITS(12) command (see table 55) defines the range within which subsequent linked commands may operate. See the SET LIMITS(10) command (6.1.16) for a description of the fields in this command.

Table 55 - SET LIMITS(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (B3h)							
1	Reserved						RdInh	WrInh
2	(MSB)							
3								
4	Logical block address							
5								(LSB)
6	(MSB)							
7								
8	Number of blocks							
9								(LSB)
10	Reserved							
11	Control							

6.2.10 UPDATE BLOCK command

The UPDATE BLOCK command (see table 56) requests that the device server logically replace data on the medium with the data sent during the DATA OUT phase.

Table 56 - UPDATE BLOCK command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (3Dh)							
1	Reserved							RelAdr
2	(MSB)							
3								
4	Logical block address							
5								(LSB)
6	Reserved							
7	Reserved							
8	Reserved							
9	Control							

See 6.1.2 for a description of the RelAdr bit and logical block address field.

One block of data is transferred during the DATA OUT phase.

NOTE 24 See the MODE Parameters (7.2) for a description of the behaviour of the UPDATE BLOCK command relative to the enable blank check (EBC) bit.

This standard does not define the result of a WRITE command issued to a block previously updated by an UPDATE BLOCK command when blank checking is disabled.

NOTE 25 It is recommended that the device server inhibit this behaviour.

A logical block may be updated until the alternate block area is exhausted. The alternate blocks used for the update operation shall not be reported in the READ CAPACITY data. If the alternate block area is exhausted, the command shall be terminated with CHECK CONDITION and the sense key shall be set to MEDIUM ERROR, and the additional sense code set to NO DEFECT SPARE LOCATION AVAILABLE.

If the report updated block read parameter specifies posting of recovered errors for a read operation of a logical block that has had a successful update operation performed, occurs the command shall terminate with a CHECK CONDITION status. The sense key shall be set to RECOVERED ERROR and the additional sense code set to UPDATED BLOCK READ.

6.2.11 VERIFY(10) command

The VERIFY command (see table 57) requests that the device server verify the data on the medium.

Table 57 - VERIFY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Fh)							
1	Reserved			DPO	Reserved	BlkVfy	BytChk	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Verification length				(LSB)
9	Control							

If the MODE SELECT command is implemented, and the Verify Error Recovery Parameters page is also implemented, then the current settings in that page define the verification criteria. If the Verify Error Recovery Parameters page is not implemented, then the verification criteria is vendor unique.

A byte check (BytChk) bit of zero causes a medium verification to be performed with no data comparison. A BytChk bit of one causes a byte- by-byte compare of the data written on the medium and the data transferred from the application client. The data shall be transferred as it would be for a WRITE command. If the compare is unsuccessful, for any reason, the device server shall return CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

A blank verify (BlkVfy) bit of one causes a verification that the blocks are blank.

If the BytChk is one when the BlkVfy bit is one, this shall be considered an error. The device server shall return CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

See 6.1.6 for a description of the cache control bits (DPO). See 6.1.2 for a description of the RelAdr bit and logical block address field.

The verification length specifies the number of contiguous logical blocks of data or blanks that shall be verified. A verification length of zero indicates that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value indicates the number of logical blocks that shall be verified.

6.2.12 VERIFY(12) command

The VERIFY(12) command (see table 58) requests that the device server verify the data on the medium. See the VERIFY(10) command (6.2.11) for a description of the fields in this command.

Table 58 - VERIFY(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (AFh)							
1	Reserved			DPO	Reserved	BlkVfy	BytChk	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	(MSB)							
7								
8				Verification length				
9								(LSB)
10	Reserved							
11	Control							

6.2.13 WRITE(10) command

The WRITE(10) command (see table 59) requests that the device server write the data transferred from the application client to the medium.

Table 59 - WRITE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Ah)							
1	Reserved			DPO	FUA	EBP	Reserved	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Transfer length				(LSB)
9	Control							

An erase by-pass (EBP) bit of zero indicates that the logical unit will default to the normal write operation. An EBP bit of one indicates that the device server is allowed to by-pass the erase operation prior to writing the data. When accessing write-once media, the EBP bit shall be considered reserved.

See 6.1.6 for a description of the cache control bits (DPO and FUA). See 6.1.2 for a description of the RelAdr bit and logical block address field.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no data shall be transferred. This condition shall not be considered an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

This command shall be terminated with a status of RESERVATION CONFLICT if any reservation access conflict (see 6.1.12) exists, and no data shall be written.

6.2.14 WRITE(12) command

The WRITE(12) command (see table 60) requests that the device server write the data transferred from the application client to the medium. See the WRITE(10) command (6.2.13) for a description of the fields in this
command.

Table 60 - WRITE(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (AAh)							
1	Reserved			DPO	FUA	EBP	Reserved	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	(MSB)							
7								
8				Transfer length				
9								(LSB)
10	Reserved							
11	Control							

6.2.15 WRITE AND VERIFY(10) command

The WRITE AND VERIFY(10) command (see table 61) requests that the device server write the data transferred from the application client to the medium and then verify that the data is correctly written.

Table 61 - WRITE AND VERIFY(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (2Eh)							
1	Reserved			DPO	Reserved	EBP	BytChk	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	Reserved							
7	(MSB)							
8				Transfer length				(LSB)
9	Control							

If the MODE SELECT command is implemented, and the Verify Error Recovery Parameters page is also implemented, then the current settings in that page define the verification criteria. If the Verify Error Recovery Parameters page is not implemented, then the verification criteria is vendor unique.

A byte check (BytChk) bit of zero causes a medium verification to be performed with no data comparison. A BytChk bit of one causes a byte- by-byte compare of the data written on the medium and the data transferred from the application client. The data shall be transferred as it would be for a WRITE command. If the compare is unsuccessful, for any reason, the device server shall return CHECK CONDITION status and the sense key shall be set to MISCOMPARE.

An erase by-pass (EBP) bit of zero indicates that the logical unit will default to the normal write operation. An EBP bit of one indicates that the device server is allowed to by-pass the erase operation prior to writing the data. When accessing write-once media, the EBP bit shall be considered reserved.

See 6.1.6 for a description of the cache control bits (DPO). See 6.1.2 for a description of the RelAdr bit and logical block address field.

The transfer length specifies the number of contiguous logical blocks of data that shall be transferred. A transfer length of zero indicates that no logical blocks shall be transferred. This condition shall not be considered as an error and no data shall be written. Any other value indicates the number of logical blocks that shall be transferred.

6.2.16 WRITE AND VERIFY(12) command

The WRITE AND VERIFY(12) command (see table 62) requests that the device server write the data transferred from the application client to the medium and then verify that the data is correctly written. See the WRITE AND VERIFY(10) command (6.2.15) for a description of the bits in this command.

Table 62 - WRITE AND VERIFY(12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (AEh)							
1	Reserved			DPO	Reserved	EBP	BytChk	RelAdr
2	(MSB)							
3								
4				Logical block address				
5								(LSB)
6	(MSB)							
7								
8				Transfer length				
9								(LSB)
10	Reserved							
11	Control							

6.3 Commands for write-once logical units

The commands for write-once logical units shall be as shown in table 63.

Table 63 - Commands for write-once logical units

Command name	Operation code	Type	Subclause
CHANGE DEFINITION	40h	O	SPC
COMPARE	39h	O	SPC
COPY	18h	O	SPC
COPY AND VERIFY	3Ah	O	SPC
INQUIRY	12h	M	SPC
LOCK-UNLOCK CACHE	36h	O	6.1.2
LOG SELECT	4Ch	O	SPC
LOG SENSE	4Dh	O	SPC
MODE SELECT(6)	15h	O	SPC
MODE SELECT(10)	55h	O	SPC
MODE SENSE(6)	1Ah	O	SPC
MODE SENSE(10)	5Ah	O	SPC
MOVE MEDIUM	A5h	O	SMC
PORT STATUS	11h	M[1]	SPC
PRE-FETCH	34h	O	6.1.3
PREVENT-ALLOW MEDIUM REMOVAL	1Eh	O	6.1.4
READ(6)	08h	O	6.1.5
READ(10)	28h	M	6.1.6
READ(12)	A8h	O	6.2.4
READ BUFFER	3Ch	O	SPC
READ CAPACITY	25h	M	6.1.7
READ ELEMENT STATUS	B8h	O	SMC
READ LONG	3Eh	O	6.1.9
REASSIGN BLOCKS	07h	O	6.1.10
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	SPC
RELEASE	17h	M	6.1.11
REQUEST SENSE	03h	M	SPC
RESERVE	16h	M	6.1.12
REZERO UNIT	01h	O	6.1.13
SEARCH DATA EQUAL (10)	31h	O	6.1.14
SEARCH DATA EQUAL (12)	B1h	O	6.2.8
SEARCH DATA HIGH (10)	30h	O	6.1.14
SEARCH DATA HIGH (12)	B0h	O	6.2.8
SEARCH DATA LOW (10)	32h	O	6.1.14
SEARCH DATA LOW (12)	B2h	O	6.2.8
SEEK(6)	0Bh	O	6.1.15
SEEK(10)	2Bh	O	6.1.15
SEND DIAGNOSTIC	1Dh	M	SPC
SET LIMITS (10)	33h	O	6.1.16
SET LIMITS (12)	B3h	O	6.2.9
START STOP UNIT	1Bh	O	6.1.17
SYNCHRONIZE CACHE	35h	O	6.1.18
TEST UNIT READY	00h	M	SPC
VERIFY (10)	2Fh	O	6.2.11
VERIFY (12)	AFh	O	6.2.12
WRITE(6)	0Ah	O	6.1.20
WRITE(10)	2Ah	M	6.2.13
WRITE(12)	AAh	O	6.2.14
WRITE AND VERIFY (10)	2Eh	O	6.2.15
WRITE AND VERIFY (12)	AEh	O	6.2.16

WRITE BUFFER	3Bh	O	SPC
WRITE LONG	3Fh	O	6.1.23
Key: M = Command implementation is mandatory.			
O = Command implementation is optional.			
SPC = SCSI-3 Primary Commands			
SMC = SCSI-3 Medium Changer Command Set			
Notes: [1] Mandatory only if the dual port option is implemented.			

The following command codes are vendor-specific: 02h, 05h, 06h, 09h, 0Ch, 0Dh, 0Eh, 0Fh, 10h, 11h, 13h, 14h, 19h, 20h, 21h, 22h, 23h, 24h, 26h, 27h, 29h, and C0h through FFh. All remaining command codes for write-once logical units are reserved for future standardization.

7 Parameters for block device logical units

7.1 Parameters for direct-access logical units

7.1.1 Diagnostic parameters

This subclause defines the descriptors and pages for diagnostic parameters used with direct-access logical units.

The diagnostic page codes for direct-access logical units are defined in table 64.

Table 64 - Diagnostic page codes

Page code	Description	Subclause
00h	Supported diagnostic pages	SPC
40h	Translate address page – SEND DIAGNOSTIC	7.1.1.1
40h	Translate address page – RECEIVE DIAGNOSTIC	7.1.1.2
41h	Drive status page – SEND DIAGNOSTIC	7.1.1.3
41h	Drive status page – RECEIVE DIAGNOSTIC	7.1.1.4
01h – 3Fh	Reserved (for all device type pages)	
42h – 7Fh	Reserved	
80h – FFh	Vendor–specific pages	

7.1.1.1 Translate address page - SEND DIAGNOSTIC

The translate address page allows the application client to translate a logical block address, physical sector address or physical bytes from index address into any one of the other formats. The address to be translated is passed to the device server with the SEND DIAGNOSTIC command and the results are returned to the application client by the RECEIVE DIAGNOSTIC RESULTS command. The format of the translate address page - SEND DIAGNOSTIC is shown in table 65. The translated address is returned in the translate address page - RECEIVE DIAGNOSTIC RESULTS (see 7.1.1.2).

Table 65 - Translate address page - SEND DIAGNOSTIC

Bit / Byte	7	6	5	4	3	2	1	0
0	Page code (40h)							
1	Reserved							
2	(MSB) Page length (000Ah)							
3								(LSB)
4	Reserved					Supplied format		
5	Reserved					Translate format		
6 – 13	Address to translate							

The supplied format field specifies the format of address to translate field. Valid values for this field are defined in the FORMAT UNIT command (see 6.1.1). If the device server does not support the requested format it shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and an additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

The translate format field specifies which format the application client would like the address to be translated to. Valid values for this field are defined in the FORMAT UNIT command (see 6.1.1). If the device server does not support the requested format it shall terminate the command with CHECK

CONDITION status. The sense key shall be set to ILLEGAL REQUEST and an additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

The address to translate field contains a single address the application client is requesting the device server to translate. The format of this field depends on the value in the supplied format field. The formats are described in 6.1.1.1. If the logical block format is specified the block address shall be in the first four bytes of the field with the remaining bytes set to zero.

7.1.1.2 Translate address page - RECEIVE DIAGNOSTIC

The translate address page allows the application client to translate a logical block address, physical sector address, or physical bytes from index address into any one of the other formats. The address to be translated is passed to the device server with the SEND DIAGNOSTIC command and the results are returned to the application client by the RECEIVE DIAGNOSTIC RESULTS command. The translated address is returned in the translate address page - RECEIVE DIAGNOSTIC (see table 66).

Table 66 - Translate address page - RECEIVE DIAGNOSTIC

Bit / Byte	7	6	5	4	3	2	1	0
0	Page code (40h)							
1	Reserved							
2	(MSB) Page length							
3								(LSB)
4	Reserved					Supplied format		
5	RAREA	ALTSEC	ALTTRK	Reserved		Translated format		
	Translated addresses							
6 – 13	Translated address 1							
	.							
n – n+7	Translated address x (if required)							

The translate address page contains a four byte page header which specifies the page code and length followed by two bytes which describe the translated address followed by zero or more translated address(s).

The page length field contains the number of parameter bytes which follow.

The supplied format field contains the value from the SEND DIAGNOSTIC command supplied format field (see 7.1.1.1).

A reserved area (RAREA) bit of one indicates that all or part of the translated address falls within a reserved area of the medium (e.g. speed tolerance gap, alternate sector, vendor reserved area, etc.). If the entire translated address falls within a reserved area the device server may not return a translated address. An RAREA bit of zero indicates that no part of the translated address falls within a reserved area of the medium.

An alternate sector (ALTSEC) bit of one indicates that the translated address is physically located in an alternate sector of the medium. If the device server cannot determine if all or part of the translated address is located in an alternate sector it shall set this bit to zero. An ALTSEC bit of zero indicates that no part of the translated address is located in an alternate sector of the medium or that the device server is unable to determine this information.

An alternate track (ALTTRK) bit of one indicates that part or all of the translated address is located on an alternate track of the medium or the device server cannot determine if all or part of the translated address is located on an alternate track. An ALTTRK bit of zero indicates that no part of the translated address is located on an alternate track of the medium.

The translated format field contains the value from the SEND DIAGNOSTIC command translate format field (see 7.1.1.1).

The translated address field contains the address(s) the device server translated from the address supplied by the application client in the SEND DIAGNOSTIC command. This field shall be in the format specified in the translate format field. The different formats are described in 6.1.1.1. If the logical block format is specified the block address shall be in the first four bytes of the field and the remaining bytes shall be set to zero.

If the returned data is in the logical block or physical sector format and the address to be translated covers more than one address after it has been translated (e.g. accounting for speed tolerance or multiple physical sectors within a single logical block or multiple logical blocks within a single physical sector) the device server shall return all possible addresses which are contained in the area specified by the address to be translated.

If the returned data is in bytes from index format the device server shall return a pair of translated values for each of the possible addresses which are contained in the area specified by the address to translate field. Of the pair of translated values returned, the first indicates the starting location and the second the ending location of the area.

7.1.1.3 Drive status page - SEND DIAGNOSTIC

The drive status page allows the application client to query the drive regarding operational status of the drive The format of the drive status page - SEND DIAGNOSTIC is shown in Table 67. The drive status information is returned in the drive status page - RECEIVE DIAGNOSTIC.

Table 67 - Drive status page - SEND DIAGNOSTIC

Bit / Byte	7	6	5	4	3	2	1	0
0	Page code (41h)							
1	Reserved							
2	(MSB)							
3	Page length (0008h)							(LSB)
4 – 7	Reserved							
8 – 11	Reserved							

7.1.1.4 Drive status page - RECEIVE DIAGNOSTIC

The drive status page allows the application client to query the drive regarding operational status of the drive The format of the drive status page - RECEIVE DIAGNOSTIC is shown in Table 68.

Table 68 - Drive status page - RECEIVE DIAGNOSTIC

Bit / Byte	7	6	5	4	3	2	1	0
0	Page code (41h)							
1	Reserved							
2	(MSB) Page length (n-4)							
3								(LSB)
4	Reserved							
5								
6	Reserved				Synchronization		RPL	
7	Reserved					SSIS	SSIE	SSSL
8	Reserved							
47								
	Vendor specific							
48	Reserved							
n								

The Synchronization Status field is used to report whether or not the spindle has synchronized with the reference signal or to report that the synchronization is in progress. The definitions of values in this field are shown in Table 69.

Table 69 - Synchronization Status

Status	Description
00b	Synchronization status reporting is not supported or the status cannot be determined
01b	Spindle is synchronized with the reference signal
10b	Spindle cannot synchronize with the reference signal or no reference signal is present
11b	Spindle is in process of synchronizing with the reference signal

If the logical unit has not been selected as a master, master control, or slave or if the reporting of synchronous status is not supported, the synchronous status shall be set to 00b.

If no reference signal is being received but the logical unit is currently a master, slave, or master control, the synchronization status will be set to 10b.

Once the reference signal is received, the logical unit shall begin its internal synchronization, attempting to match the drive's spindle speed to the reference signal. During this time, the synchronization status shall be set to 11b. The amount of time required to achieve synchronization is not defined by this standard.

If the logical unit is unable to synchronize to the reference signal, the logical unit shall set the synchronization status to 10b. The sense key shall be set to UNIT ATTENTION, the additional sense code shall be set to RPL STATUS CHANGE.

Once the logical unit successfully synchronizes to the reference signal, if the synchronization signal is lost due to a loss of the reference signal or a malfunction, unit attention conditions shall be generated for all application clients and the synchronization status shall be set to 10b. The sense key shall be set to UNIT ATTENTION. If the logical unit has successfully achieved synchronization and then loses synchronization while executing an I/O process and no other error occurs, then the device server shall return CHECK CONDITION status. The sense key shall be set to RECOVERED ERROR if the logical unit is able to complete the I/O process or HARDWARE ERROR if the logical unit is unable to complete the I/O process. The additional sense code is set to RPL STATUS CHANGE for these cases. The rotational position locking (RPL) field shall be returned as the current value found in the Rigid Disk Geometry Page (Table 91).

A synchronous spindle invalid signal (SSIS) bit of one indicates that the synchronous spindle signal received by the drive was invalid or not recognized by the drive as a valid synchronization signal. An example of this event is the receipt of synchronous signals from multiple masters. An SSIS bit of zero indicates no signal is present or a valid synchronization signal is being received.

A synchronous spindle internal error (SSIE) bit of one indicates that the synchronization spindle electronics has detected an internal failure and the spindle is not synchronized with the synchronization signal. An SSIE bit of zero indicates no internal electronic failure has been detected by the drive.

A synchronization spindle signal loss (SSSL) bit of one indicates that the drive detects receiving no synchronization signal. An SSSL bit of zero indicates that a spindle synchronization signal is being received.

The above three synchronization status error reporting fields, SSIS, SSIE, and SSSL, are used to indicate error conditions of a drive set to master or slave spindle synchronization mode using the Rigid Disk Geometry Mode Select page RPL field. A drive not set as master or slave shall report zero in these fields.

7.1.2 Log parameters

This subclause defines the descriptors and pages for log parameters used with direct-access logical units. See SPC for a detailed description of Logging Operations.

The log page codes for direct-access logical units are defined in table 70.

Table 70 - Log page codes

Page code	Description	Subclause
01h	Buffer over-run/under-run page	SPC
03h	Error counter page (read) page	SPC
04h	Error counter page (read reverse) page	SPC
05h	Error counter page (verify) page	SPC
02h	Error counter page (write) page	SPC
07h	Last n error events page	SPC
06h	Non-medium error page	SPC
00h	Supported log pages	SPC
08h	Format status page	7.1.2.1
09h - 2Fh	Reserved	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

7.1.2.1 Format status page

This page (page code 08h) captures the state of the logical unit since the most recent successful FORMAT UNIT command performed. Additionally, this page records the impact that Defect Management has had on the logical unit.

Table 71 defines the parameter codes for the Format Status Log Page.

Table 71 - Format Status Page

Paramater Code	Description
0000h	Format DATA OUT
0001h	Grown defects during certification
0002h	Total blocks reallocated during format
0003h	Total new blocks reallocated
0004h	Power on minutes since format
0005h - 7FFFh	Reserved
8000h - FFFFh	Vendor-specific pages

Event counts are returned as a result of the LOG SENSE command. LOG SELECT cannot be used to preset (a value other than zero) any of the above event counts. Attempts to change these event counts by issuing a LOG SELECT with these fields set to non-zero values is not considered an error and shall have no effect on the saved values.

All of the log parameters described above shall be reported as -1 (all FFh in all bytes of the log parameter) if the most recent FORMAT UNIT command failed. Individual log parameters described above shall be reported as -1 (all FFh in all bytes of the log parameter) if no such information is available.

Format DATA OUT: This field contains the entire DATA OUT phase of the most recently successful FORMAT UNIT operation performed. This includes the Defect List Header (4 bytes), the Initialization Pattern Descriptor(s) if any (variable number of bytes), and the Defect Descriptor(s) if any (variable number of bytes). Refer to section 6.1.1 for details about these fields.

Grown Defects During Certification: This is a counter of the number of defects detected as a result of performing Certification during execution of a FORMAT UNIT command. This count reflects only those

defects detected and replaced which were not already part of the Plist or Glist. If a Certification pass was not performed this field shall be returned as zero.

Total Blocks Reallocated During Format: This is a counter of the total number of blocks which have been reallocated since the completion of the last successful FORMAT UNIT command as of the current LOG SENSE request.

Power On Minutes Since Format: This field represents the unsigned number of usage minutes (power applied regardless of power state) which have elapsed since the most recently successful FORMAT UNIT command.

NOTE 26 Upon receiving the FORMAT UNIT command, the device server should set all fields within the Format Status Log Page to reflect no such information being available. Only upon successful completion of the FORMAT UNIT command should the device server update the affected fields.

NOTE 27 The TSD (Target Save Disable) bit is always returned as 0 to indicate that the device server shall provide an implicit saving frequency so that the information contained in the log page has statistical significance.

NOTE 28 The intent relative to removable media is that log page information be stored with the media in a vendor specific manner and location.

7.1.3 Mode parameters

This subclause defines the descriptors and pages for mode parameters used with direct-access logical units.

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

The medium-type code field is contained in the mode parameter header (see the SPC). Table 72 defines this field for direct-access logical units.

Table 72 - Direct-access medium-type codes

Code value	Medium type					
00h	Default medium type (currently mounted medium type)					
01h	Flexible disk, single-sided; unspecified medium					
02h	Flexible disk, double-sided; unspecified medium					
	Flexible disks					
	Diameter mm (in)	Bit density Bits/radian	Track density /mm (/in)	Number of sides	ANSI reference standard	
05h	200 (8,0)	6 631	1,9 (48)	1	ANSI X3.73-1980	
06h	200 (8,0)	6 631	1,9 (48)	2	(Note 1)	
09h	200 (8,0)	13 262	1,9 (48)	1	None	
0Ah	200 (8,0)	13 262	1,9 (48)	2	ANSI X3.121-1984	
0Dh	130 (5,25)	3 979	1,9 (48)	1	ANSI X3.82-1980	
12h	130 (5,25)	7 958	1,9 (48)	2	ANSI X3.125-1985	
16h	130 (5,25)	7 958	3,8 (96)	2	ANSI X3.126-1986	
1Ah	130 (5,25)	13 262	3,8 (96)	2	ISO DIS8630-1985	
1Eh	90 (3,5)	7 958	5,3 (135)	2	ANSI X3.137 (Note 2)	
	Direct-access magnetic yapes					
	Width mm (in)	Tracks	Density ftpmm (ftpi)	ANSI reference standard		
40h	6,3 (0,25)	12	394 (10 000)	Note 1		
44h	6,3 (0,25)	24	394 (10 000)	Note 1		
Code values 80h - FFh are vendor-specific. All remaining code values are reserved.						
NOTES 1 See annex c for additional standards information. 2 This standard is under development. Please contact the secretariat for current information on status and availability.						

The device specific parameter field (see table 73) is contained in the mode parameter header (see the SPC).

Table 73 - Device specific parameter

Bit	7	6	5	4	3	2	1	0
	WP	Reserved		DPOFUA	Reserved			

When used with the MODE SELECT command the write protect (WP) bit is not defined.

When used with the MODE SENSE command a WP bit of zero indicates that the medium is write enabled. A WP bit of one indicates that the medium is write protected.

When used with the MODE SELECT command, the DPOFUA bit is not used and the field is reserved.

When used with the MODE SENSE command, a DPOFUA bit of one indicates that the device server supports the DPO and FUA bits (see 6.1.6). When used with the MODE SENSE command, a DPOFUA bit of zero indicates that the device server does not support the DPO and FUA bits.

The density code field is contained in the mode parameter block descriptor (see 7.3.3). This field is reserved for direct-access logical units.

The mode page codes for direct-access logical units are shown in table 74.

Table 74 - Mode page codes

Page code	Description	Subclause
08h	Caching page	7.1.3.1
0Ah	Control mode page	SPC
02h	Disconnect-reconnect page	SPC
05h	Flexible disk page	7.1.3.2
03h	Format device page	7.1.3.3
1Ch	Informational exceptions control page	SPC
0Bh	Medium types supported page	7.1.3.4
0Ch	Notch and partition page	7.1.3.5
09h	Peripheral device page	SPC
0Dh	Power condition page	7.1.3.6
01h	Read-write error recovery page	7.1.3.6
04h	Rigid disk geometry page	7.1.3.7
07h	Verify error recovery page	7.1.3.8
06h	Reserved	
0Eh - 1Bh	Reserved	
1Dh - 1Fh	Reserved	
00h	Vendor-specific (does not require page format)	
20h - 3Eh	Vendor-specific (page format required)	
3Fh	Return all pages (valid only for the MODE SENSE command)	

For direct-access logical units, if the notch page is not supported, or if the active notch field in the notch page is zero then each page descriptor specifies mode parameters for the device server to use for subsequent operations on the specified logical unit. If the notch page is supported, and the active notch is not zero, then each page descriptor specifies parameters for the device server to use for subsequent operations on the disk notch specified by the current value of the active notch field on the specified logical unit.

7.1.3.1 Caching page

The caching parameters page (see table 75) defines the parameters that affect the use of the cache.

Table 75 - Caching page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (08h)					
1	Page length (12h)							
2	IC	ABPF	CAP	DISC	SIZE	WCE	MF	RCD
3	Demand read retention priority				Write retention priority			
4	(MSB)							
5	Disable pre-fetch transfer length							(LSB)
6	(MSB)							
7	Minimum pre-fetch							(LSB)
8	(MSB)							
9	Maximum pre-fetch							(LSB)
10	(MSB)							
11	Maximum pre-fetch ceiling							(LSB)
12	FSW	LBCSS	DRA	VS	VS	Reserved		
13	Number of cache segments							
14	(MSB)							
15	Cache Segment Size							(LSB)
16	Reserved							
17	(MSB)							
18	Non Cache Segment Size							
19								(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 device server is capable of saving the page in a non-volatile vendor-specific location. If the PS is set to one in MODE SENSE data then the page shall be savable by issuing a MODE SELECT command with SP set to one.

The Initiator Control (IC) enable bit, when set to one, requests that the device server use the number of Cache Segments or Cache Segment Size fields, dependent upon the Size bit, to control the caching algorithm rather than the device server's own adaptive algorithm.
The Abort Pre-Fetch (ABPF) bit, when set to one, with the DRA bit equal to zero, requests that the device server abort the pre-fetch upon selection. The ABPF set to one takes precedence over the Minimum Pre-fetch bytes. When set to zero, with the DRA bit equal to zero, the termination of any active pre-fetch is dependent upon Caching Page bytes 4 through 11 and is operation and/or vendor specific.

The Caching Analysis Permitted (CAP) bit, when set to one, requests that the device server perform caching analysis during subsequent operations. When set to zero, CAP requests that caching analysis be disabled to reduce overhead time or to prevent non pertinent operations from impacting tuning values.

The Discontinuity (DISC) bit, when set to one, requests that the device server continue the pre-fetch across time discontinuities, such as across cylinders (or tracks in an embedded servo drive), up to the limits of the buffer, or segment, space available for pre-fetch. When set to zero, the DISC requests that pre-fetches be truncated (or wrapped) at time discontinuities.

The Size Enable (SIZE) bit, when set to one, indicates that the Cache Segment Size is to be used to control caching segmentation. When SIZE equals zero, the application client requests that the Number of Cache Segments is to be used to control caching segmentation. Simultaneous use of both number of segments and segment size is vendor specific.

A write cache enable (WCE) bit of zero specifies that the device server shall return GOOD status for a WRITE command after successfully writing all of the data to the medium. A WCE bit of one specifies that the device server may return GOOD status for a WRITE command after successfully receiving the data and prior to having successfully written it to the medium.

A multiplication factor (MF) bit of zero specifies that the device server shall interpret the minimum and maximum pre-fetch fields in terms of the number of logical blocks for each of the respective types of pre-fetch. An MF bit of one specifies that the device server shall interpret the minimum and maximum pre-fetch fields to be specified in terms of a scaler number which, when multiplied by the number of logical blocks to be transferred for the current command, yields the number of logical blocks for each of the respective types of pre-fetch.

A read cache disable (RCD) bit of zero specifies that the device server may return data requested by a READ command by accessing either the cache or media. A RCD bit of one specifies that the device server shall transfer all of the data requested by a READ command from the medium (i.e. data cannot be transferred from the cache).

The demand read retention priority field (see table 76) advises the device server on the retention priority to assign data read into the cache that has also been transferred from the logical unit to the application client.

Table 76 - Demand read and write retention priority

Value	Description
0h	Indicates the device server should not distinguish between retaining the indicated data and data placed into the cache memory by other means (e.g. pre-fetch).
1h	Demand read retention priority: Data put into the cache via a READ command should be replaced sooner (has lower priority) than data placed into the cache by other means (e.g. pre-fetch). Write retention priority: Data put into the cache during a WRITE or WRITE AND VERIFY command should be replaced sooner (has lower priority) than data placed into the cache by other means (e.g. pre-fetch).
Fh	Demand read retention priority: Data put into the cache via a READ command should not be replaced if there is other data in the cache that was placed into the cache by other means (e.g. pre-fetch) and it may be replaced (i.e. it is not locked). Write retention priority: Data put into the cache during a WRITE or WRITE AND VERIFY command should not be replaced if there is other data in the cache that was placed into the cache by other means (e.g. pre-fetch) and it may be replaced (i.e. it is not locked).
2h - Eh	Reserved

The write retention priority field (see table 76) advises the device server on the retention priority to assign data written into the cache that has also been transferred from the cache memory to the medium.

An anticipatory pre-fetch occurs when data is placed in the cache that has not been requested. This usually happens in conjunction with the reading of data that has been requested. All the following parameters give an indication to the device server how it should manage the cache based on the last READ command. An anticipatory pre-fetch may occur based on other information. All the remaining caching parameters are only recommendations to the device server and should not cause a CHECK CONDITION to occur if the device server cannot satisfy the request.

The disable pre-fetch transfer length field specifies the selective disabling of anticipatory pre-fetch on long transfer lengths. The value in this field is compared to the number of blocks requested by the current READ command. If the number of blocks is greater than the disable pre- fetch transfer length, then an anticipatory pre-fetch is not done for the command. Otherwise the device server should attempt an anticipatory pre-fetch. If the pre-fetch disable transfer length is set to zero, then all anticipatory pre-fetching is disabled for any request for data, including those for zero logical blocks.

The minimum pre-fetch field indicates either a number of blocks or a scaler multiplier of the transfer length, depending upon the setting of the MF bit. In either case, the resulting number of blocks is the number to pre-fetch regardless of the delays it might cause in executing subsequent commands.

The pre-fetching operation begins at the logical block immediately after the last logical block of the previous READ command. Pre-fetching shall always halt before the end of the media. Errors that occur during the pre-fetching operation shall not be reported to the application client unless that device server cannot, as a result of the error, execute subsequent commands correctly. In this case the error may be reported either immediately as an error for the current READ command, or as a deferred error, at the discretion of the device server and according to the rules for reporting deferred errors.

If pre-fetch has read more than the amount of data indicated by the minimum pre-fetch then pre-fetching should be terminated whenever another command is ready to execute. This consideration is ignored when the minimum pre-fetch is equal to the maximum pre-fetch.

The maximum pre-fetch field indicates either a number of blocks or a scaler multiplier of the transfer length, depending upon the setting of the MF bit. In either case, the resulting number of blocks is the number to pre-fetch if there are no delays in executing subsequent commands.

The maximum pre-fetch field contains the maximum amount of data to pre- fetch into the cache as a result of one READ command. It is used in conjunction with the disable pre-fetch transfer length and maximum pre- fetch ceiling parameters to trade off pre-fetching new data with displacing old data already stored in the cache.

The maximum pre-fetch ceiling field specifies an upper limit on the number of logical blocks computed as the maximum pre-fetch. If this number of blocks is greater than the maximum pre-fetch, then the number of logical blocks to pre-fetch shall be truncated to the value stored in the maximum pre-fetch ceiling field.

> **NOTE 28** If the MF bit is one the maximum pre-fetch ceiling field is useful in limiting the amount of data to be pre-fetched.

The Force Sequential Write (FSW) bit when set to one, indicates that multiple block writes are to be transferred over the SCSI bus and written to the media in an ascending, sequential, logical block order. When the FSW bit equals zero, the device server is allowed to reorder the sequence of writing addressed logical blocks in order to achieve a faster command completion.

The Logical Block Cache Segment Size (LBCSS) bit when set to one, indicates that the Cache Segment Size field units shall be interpreted as logical blocks. When the LBCSS bit equals zero the Cache Segment Size field units shall be interpreted as bytes. The LBCSS shall not impact the units of other fields.

The Disable Read-Ahead (DRA) bit, when set to one, requests that the device server not read into the buffer any logical blocks beyond the addressed logical block(s). When the DRA bit equals zero, the device server may continue to read logical blocks into the buffer beyond the addressed logical block(s).

The Vendor Specific (VS) bits may optionally be used for vendor specific purposes in the Caching Page.

The Number of Cache Segments advises the device server how many segments the host requests that the cache be divided into.

The Cache Segment Size field indicates the requested segment size in Bytes. This standard assumes that the Cache Segment Size field is valid only when the SIZE bit is one.

If the Non Cache Buffer Size field is greater than zero, this field advises the device server how many bytes the application client requests that the device server allocate for a buffer function when all other cache segments are occupied by data to be retained. If the number is at least one, caching functions in the other segments need not be impacted by cache misses to perform the SCSI buffer function. The impact of the Non Cache Buffer Size equal 0 or the sum of this field plus the Cache Segment Size greater than the buffer size is vendor specific.

7.1.3.2 Flexible disk page

The flexible disk page (see table 77) contains parameters for control and reporting of flexible disk drive parameters.

Table 77 - Flexible disk page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (05h)					
1	Page length in bytes (1Eh)							
2	(MSB) Transfer rate							
3								(LSB)
4	Number of heads							
5	Sectors per track							
6	(MSB) Data bytes per sector							
7								(LSB)
8	(MSB) Number of cylinders							
9								(LSB)
10	(MSB) Starting cylinder-write precompensation							
11								(LSB)
12	(MSB) Starting cylinder-reduced write current							
13								(LSB)
14	(MSB) Drive step rate							
15								(LSB)
16	Drive step pulse width							
17	(MSB) Head settle delay							
18								(LSB)
19	Motor on delay							
20	Motor off delay							
21	TRDY	SSN	MO	Reserved				
22	Reserved					SPC		
23	Write compensation							
24	Head load delay							
25	Head unload delay							
26	Pin 34					Pin 2		
27	Pin 4					Pin 1		
28	Medium rotation rate							
29								(LSB)
30	Reserved							
31	Reserved							

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

NOTE 29 This page is mainly intended for defining parameters of flexible disk drives, but may be used for other logical units, if applicable.

The transfer rate indicates the data rate of the peripheral logical unit. See table 78 for examples of common transfer rates.

Table 78 - Examples of transfer rates

Value	Transfer rate
00FAh	250 kbit/s transfer rate
012Ch	300 kbit/s transfer rate
01F4h	500 kbit/s tranfer rate
03E8h	1 mbit/s transfer rate
07D0h	2 mbit/s transfer rate
1388h	5 mbit/s transfer rate

The number of heads field specifies the number of heads used for reading and writing data on the medium. Heads used exclusively for servo information are excluded.

The sectors per track field specifies the number of sectors per revolution per head.

The data bytes per sector field specifies the number of bytes of data per sector that an application client can read or write.

The number of cylinders field specifies the number of cylinders used for data storage.

The starting cylinder for write precompensation field specifies the cylinder at which write precompensation is to begin. Cylinders are numbered starting with zero. If the starting cylinder for write precompensation is equal to the value in the number of cylinders field, write precompensation shall be disabled by the device server.

The starting cylinder for reduced write current field specifies cylinder at which write current is reduced. Cylinders are numbered starting with zero. If the starting cylinder for reduced write current is equal to the value in the number of cylinders field, reduced write current shall be disabled by the device server.

The drive step rate field specifies the step rate in units of 100 us. This value may be rounded as defined in 7.5.4. A value of zero requests the device server to set its default value.

The drive step pulse width field specifies the width of the step pulse in microseconds. This value may be rounded as defined in 7.5.4. A value of zero requests the device server to set its default value.

The head settle delay field specifies the head settle time in units of 100 us. This value may be rounded as defined in 7.5.4. A value of zero requests the device server to set its default value.

If a true ready signal is not available, the motor on delay field specifies in tenths of a second the time that the device server shall wait before attempting to access the medium after the motor on signal is asserted. If a true ready signal is available, the motor on delay field specifies in tenths of a second the time that the device server shall wait for drive ready status before aborting an attempt to access the medium. This value may be rounded as defined in 7.5.4.

The motor off delay field specifies in tenths of a second the time that the device server shall wait before releasing the motor on signal after an idle condition exists. A value of FFh indicates that the motor on

signal shall not be released. The START STOP UNIT command is not affected by this parameter. This value may be rounded as defined in 7.5.4.

A true ready (TRDY) bit of one specifies that a signal is provided that indicates the medium is ready to be accessed.

A start sector number (SSN) bit of one specifies that sectors are numbered starting with one. An SSN bit of zero specifies that sectors are numbered starting with zero.

A motor on (MO) bit of one specifies that pin 16 (motor on) shall remain released. This bit shall be set to one when using high capacity (192 tracks per inch) drives and their pre-formatted diskettes. An MO bit of zero indicates that pin 16 (motor on) shall be asserted.

The step pulse per cylinder (SPC) field is used to specify the number of additional step pulses required per cylinder. Non-zero values allow a drive to read a diskette formatted on a drive with a lower number of tracks per inch. For example, a value of one allows a 96 track-per-inch drive to access tracks on a diskette that was formatted for 48 tracks per inch.

The write compensation field is used to specify the amount of write compensation to be used starting at the cylinder specified in the starting cylinder for write precompensation field. The correlation of any values used in this field to actual write precompensation time values is vendor- specific. If a zero is specified in this field the device server shall use its default write precompensation value. This value may be rounded as defined in 7.5.4.

The head load delay field specifies the head loading time in milliseconds. This value may be rounded as defined in 7.5.4. A value of zero requests the device server to set its default value.

The head unload delay field specifies the head unloading time in milliseconds. This value may be rounded as defined in 7.5.4. A value of zero requests the device server to set its default value.

The Pin 34 field defines the usage of pin 34 of the flexible disk drive interface. This use of this pin varies among vendors and drives. The settings allow the application client to select how pin 34 shall be used by the interface. See table 79.

Table 79 - Pin 34 field

Bit 7 6 5 4	Description of pin 34 use
P 0 0 0	Open
P 0 0 1	Ready
P 0 1 0	Disk changed

NOTES
1 P is a polarity bit, where 0 is active low and 1 is active high.
2 All undefined values are reserved.

The pin 4 field defines the usage of pin 4 of the flexible disk drive interface. This use of this pin varies among drive vendors and drives. The settings allow the application client to specify how pin 4 shall be used by the interface. See table 80.

Table 80 - Pin 4 field

Bit 7 6 5 4	Description of pin 4 use
P 0 0 0	Open
P 0 0 1	In use
P 0 1 0	Eject
P 0 0 0	Head load

NOTES
1 P is a polarity bit, where 0 is active low and 1 is active high.
2 All undefined values are reserved.

The pin 1 field defines the usage of pin 1 of the flexible disk drive interface. This use of this pin varies among vendors and drives. The settings allow the application client to specify how pin 1 shall be used by the interface. See table 81.

Table 81 - Pin 34 field

Bit 7 6 5 4	Description of pin 34 use
P 0 0 0	Open
P 0 0 1	Disk change reset

NOTES
1 P is a polarity bit, where 0 is active low and 1 is active high.
2 All undefined values are reserved.

The medium rotation rate field specifies the speed at which the medium rotates. The unit of measure is rotations per minute (e.g. 2 400 rpm). This field cannot be changed by a MODE SELECT command.

7.1.3.3 Format device page

The format device page (see table 82) contains parameters which specify the medium format.

Table 82 - Format device page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (03h)					
1	Page length (16h)							
2	(MSB)							
3	Tracks per zone							(LSB)
4	(MSB)							
5	Alternate sectors per zone							(LSB)
6	(MSB)							
7	Alternate tracks per zone							(LSB)
8	(MSB)							
9	Alternate tracks per logical unit							(LSB)
10	(MSB)							
11	Sectors per track							(LSB)
12	(MSB)							
13	Data bytes per physical sector							(LSB)
14	(MSB)							
15	Interleave							(LSB)
16	(MSB)							
17	Track skew factor							(LSB)
18	(MSB)							
19	Cylinder skew factor							(LSB)
20	SSEC	HSEC	RMB	SURF	Reserved			
21 - 23	Reserved							

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

> **NOTE 30** If the application client changes any of the current physical parameters defined below, the device server may not be able to access the media until a FORMAT UNIT command has been successfully completed.

If the defect handling format parameters (tracks per zone, alternate sectors per zone, alternate tracks per zone and alternate tracks per logical unit) requested by the application client are not supported by the device server the device server may round these fields to acceptable values as described in 7.5.4.

The tracks per zone field specifies the number of tracks per zone to use in dividing the capacity of the logical unit for the purpose of allocating alternate sectors. A value of zero means that one zone is

defined for the entire logical unit. The last zone on the logical unit might not contain the same number of tracks as the previous zone(s).

The alternate sectors per zone field specifies the number of sectors per zone the device server shall reserve for defect handling. The device server shall de- allocate these sectors from the application client addressable blocks during the FORMAT UNIT command. If the notch page is implemented and the ND bit of the notch page is one and the active notch field of the notch page is zero, then a value of zero indicates that no alternate sectors shall be reserved. Otherwise, a value of zero indicates that the number of alternate sectors is logical unit specific.

The alternate tracks per zone field specifies the number of tracks per zone the device server shall reserve for defect handling. The device server shall de- allocate these tracks from the application client addressable blocks during the FORMAT UNIT command. If the notch page is implemented and the ND bit of the notch page is one and the active notch field of the notch page is zero, then a value of zero indicates that no alternate tracks shall be reserved. Otherwise, a value of zero indicates that the number of alternate tracks is logical unit specific.

The alternate tracks per logical unit field specifies the number of tracks per logical unit the device server shall reserve for defect handling. The device server shall de-allocate these tracks from the application client addressable blocks during the FORMAT UNIT command. If the notch page is implemented and the ND bit of the notch page is one and the active notch field of the notch page is zero, then a value of zero indicates that no alternate tracks shall be reserved. Otherwise, a value of zero indicates that the number of alternate tracks is logical unit specific.

The sectors per track field specifies the number of physical sectors included within each track. This number includes any alternate sectors the device server may allocate. A value of zero during MODE SELECT indicates that the device server shall define the number of sectors per track. For logical units with a variable number of sectors per track, the value in MODE SELECT shall be zero and the value reported in MODE SENSE for the number of sectors per track is vendor-specific.

The data bytes per physical sector field specifies the number of data bytes per physical sector that the device server shall use. This value may be different than the logical block size reported in the MODE SELECT data. The device server shall return CHECK CONDITION status if it determines that the combination of this field and the sectors per track field exceed the capability of the medium. A value of zero indicates that the data bytes per physical sector is defined by the logical unit.

For MODE SENSE the interleave field returns the same parameter passed in the FORMAT UNIT command, The device server shall report this field as defined in the corresponding MODE SENSE command. For MODE SELECT this field is ignored.

> **NOTE 31** It is recommended that this field be marked non-changeable and that application clients send the value returned in MODE SENSE. This allows migration to specifying interleave as a mode parameter instead of in the FORMAT UNIT command.

The track skew factor field specifies the number of physical sectors between the last logical block of one track and the first logical block on the next sequential track of the same cylinder.

The cylinder skew factor field specifies the number of physical sectors between the last logical block of one cylinder and the first logical block on the next sequential cylinder.

The SSEC bit set to one indicates that the device server shall use soft sector formatting.

The HSEC bit set to one indicates that the device server shall use hard sector formatting. The HSEC bit and the SSEC bit are mutually exclusive in MODE SELECT commands.

The combinations sector formatting supported that are reported in response to a request for default values are defined in table 83.

Table 83 - Reporting of default sector formatting support

SSEC	HSEC	Description
0	0	Device server shall not return this combination
1	0	Device server supports soft sector formatting only
0	1	Device server supports hard sector formatting only
1	1	Device server supports both soft and hard sector formatting

The combinations sector formatting supported that are reported in response to a request for changeable values are defined in table 84.

Table 84 - Reporting of changeable sector formatting support

SSEC	HSEC	Description
0	0	Sector formatting not changeable
1	0	Device server shall not return this combination
0	1	Device server shall not return this combination
1	1	Device server supports both soft and hard sector formatting

The removable (RMB) bit set to one indicates that the logical unit supports removable media. A RMB bit set to zero indicates that the logical unit does not support removable media. The status of this bit shall be reflected in the INQUIRY command removable media bit (RMB).

The surface (SURF) bit set to zero indicates that the device server shall allocate progressive addresses to all logical blocks within a cylinder prior to allocating addresses on the next cylinder. A SURF bit set to one indicates that the device server shall allocate progressive addresses to all logical blocks on a surface prior to allocating sector addresses on the next surface.

> **NOTE 32** If the device server supports savable parameters, all savable parameters for this application client, including those in page codes 3, 4, and 5, are saved to non-volatile memory when the save parameters bit (SP) in the command descriptor block is set to one. The savable parameters may also be saved to non-volatile memory during a FORMAT UNIT command (see 6.1.1).

7.1.3.4 Medium types supported page

The medium types supported page (see table 85) contains a list of the medium types implemented by the device server for logical units.

Table 85 - Medium types supported page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved		Page code (0Bh)				
1				Page length (06h)				
2				Reserved				
3				Reserved				
4				Medium type one supported				
5				Medium type two supported				
6				Medium type three supported				
7				Medium type four supported				

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 device server is capable of saving the page in a non-volatile vendor-specific location.

The code values for each medium type supported by the device server (up to four maximum), are reported in ascending order. If only the default medium type is supported zero is reported. If less than four medium types are supported the unused entries shall be returned as zero.

7.1.3.5 Notch and partition page

The notch page (see table 86) contains parameters for direct-access logical units which implement a variable number of blocks per cylinder and support this page. Each clause of the logical unit with a different number of blocks per cylinder is referred to as a notch.

Table 86 - Notch page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved		Page code (0Ch)				
1				Page length (16h)				
2	ND	LPN		Reserved				
3				Reserved				
4	(MSB)			Maximum number of notches				
5								(LSB)
6	(MSB)			Active notch				
7								(LSB)
8	(MSB)			Starting boundary				
11								(LSB)
12	(MSB)			Ending boundary				
15								(LSB)
16	(MSB)			Pages notched				
23								(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 device server is capable of saving the page in a non-volatile vendor-specific location.

A notched drive (ND) bit of zero indicates that the logical unit is not notched and that all other parameters in this page shall be returned as zero by the device server. A ND bit of one indicates that the logical unit is notched. For each supported active notch value this page defines the starting and ending boundaries of the notch.

A logical or physical notch (LPN) bit of zero indicates that the boundaries are based on the physical parameters of the logical unit. The cylinder is considered most significant, the head least significant. A LPN bit of one indicates that the notch boundaries are based on logical blocks of the logical unit.

The maximum number of notches field indicates the maximum number of notches supported by the logical unit. This field shall be reported as unchangeable.

The active notch field indicates the notch that this and subsequent MODE SELECT and MODE SENSE commands shall refer to, until the active notch is changed by a later MODE SELECT command. The value of the active notch shall be greater than or equal to 0 and less than or equal to the maximum number of notches. An active notch value of zero indicates that this and subsequent MODE SELECT and MODE SENSE commands refer to the parameters that apply across all notches.

The starting boundary field indicates the beginning of the active notch or, if the active notch is zero, the beginning boundary of the logical unit. If the LPN bit is one, then the four bytes represent a logical block address. If the LPN bit is zero, then the three most significant bytes shall represent the cylinder number

and the least significant byte shall represent the head number. This field shall be reported as unchangeable. When used with the MODE SELECT command this field is ignored.

The ending boundary field indicates the ending of the active notch or, if the active notch is zero, the ending of the logical unit. If the LPN bit is one, then the four bytes represent logical block address. If the LPN bit is zero, then the three most significant bytes shall represent the cylinder number and the least significant byte shall represent the head number. This field shall be reported as unchangeable. When used with the MODE SELECT command this field is ignored.

Each notch shall span a set of consecutive logical blocks on the logical unit, the notches shall not overlap, and no logical block shall be excluded from a notch.

The pages notched field is a bit map of the mode page codes that indicates which pages contain parameters that may be different for different notches. The most significant bit of this field corresponds to page code 3Fh and the least significant bit corresponds to page code 00h. If a bit is one, then the corresponding mode page contains parameters that may be different for different notches. If a bit is zero, then the corresponding mode page contains parameters that are constant for all notches. This field shall be reported as unchangeable.

7.1.3.6 Power condition page

The power condition page (table 87) provides the application client the means to control the length of time a logical unit will delay before changing its power requirements. There is no notification to the application client that a logical unit has entered into one of the power conditions.

On the receipt of a command the device shall adjust itself to the power condition which allows the command to execute. The timer which maps to this power condition and any lower power condition timers shall be reset on receipt of the command. On completion of the command the timer associated with this power condition shall be restarted.

Table 87 - Power condition page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (0Dh)					
1	Page length (0Ah)							
2	Reserved							
3	Reserved						Idle	Standby
4	(MSB)							
5								
6	Idle Condition Timer							
7								(LSB)
8	(MSB)							
9								
10	Standby Condition Timer							
11								(LSB)

An Idle bit of one indicates a logical unit shall use the Idle Condition Timer to determine the length of inactivity time to wait before entering the Idle condition. An idle bit of zero indicates a logical unit shall not enter the Idle condition.

A Standby bit of one indicates a logical unit shall use the Standby Condition Timer to determine the length of inactivity time to wait before entering the Standby condition. A standby bit of zero indicates a logical unit shall not enter the Standby condition.

The Idle Condition Timer field indicates the inactivity time in 100 millisecond increments that the logical unit shall wait before entering the Idle condition.

If the Idle bit is one, a value of zero in the Idle Condition Timer indicates the logical unit shall enter the Idle condition on completion of any command.

The Standby condition Timer field indicates the inactivity time in 100 millisecond increments that the logical unit shall wait before entering the Standby condition. This timer shall only count if the Idle condition Timer is equal to zero.

If the Standby bit is one and the Idle bit is zero, a value of zero in the Standby Condition Timer indicates the logical unit shall enter the standby condition on completion of any command.

If the Standby bit is one and the Idle bit is one, a value of zero in the Standby Condition Timer indicates the logical unit shall enter the Standby condition when the Idle Condition Timer equals zero.

Figure 2 shows graphically the relationships between the different power conditions and their timers.

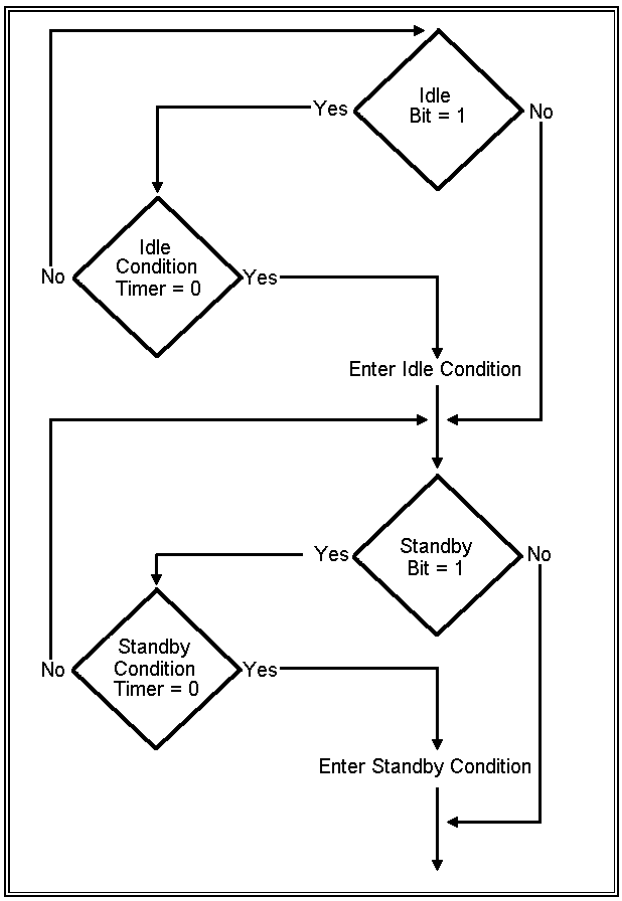

Figure 2 - Power conditions flowchart

7.1.3.6 Read-write error recovery page

The read-write error recovery page (see table 88) specifies the error recovery parameters the device server shall use during any command that performs a read or write operation to the medium (e.g. READ(6), READ(10), WRITE(6), WRITE(10), COPY, COMPARE, WRITE & VERIFY, etc.).

Table 88 - Read-write error recovery page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (01h)					
1	Page length (0Ah)							
2	AWRE	ARRE	TB	RC	EER	PER	DTE	DCR
3	Read retry count							
4	Correction span							
5	Head offset count							
6	Data strobe offset count							
7	Reserved							
8	Write retry count							
9	Reserved							
10	(MSB) Recovery time limit							
11	(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 device server is capable of saving the page in a non-volatile vendor-specific location.

An automatic write reallocation enabled (AWRE) bit of one indicates that the device server shall enable automatic reallocation to be performed during write operations. The automatic reallocation shall be performed only if the device server has the valid data (e.g. original data in the buffer or recovered from the medium). The valid data shall be placed in the reallocated block. Error reporting as required by the error recovery bits (EER, PER, DTE, and DCR) shall be performed only after completion of the reallocation. The reallocation operation shall report any failures that occur. See the REASSIGN BLOCKS command (6.1.10) for error procedures.

An AWRE bit of zero indicates that the device server shall not perform automatic reallocation of defective data blocks during write operations.

An automatic read reallocation enabled (ARRE) bit of one indicates that the device server shall enable automatic reallocation of defective data blocks during read operations. All error recovery actions required by the error recovery bits (TB, EER, PER, DTE, and DCR) shall be executed. The automatic reallocation shall then be performed only if the device server successfully recovers the data. The recovered data shall be placed in the reallocated block. Error reporting as required by the error recovery bits shall be performed only after completion of the reallocation. The reallocation process shall present any failures that occur. See the REASSIGN BLOCKS command (6.1.10) for error procedures.

An ARRE bit of zero indicates that the device server shall not perform automatic reallocation of defective data blocks during read operations.

A transfer block (TB) bit of one indicates that a data block that is not recovered within the recovery limits specified shall be transferred to the application client before CHECK CONDITION status is returned. A TB bit of zero indicates that such a data block shall not be transferred to the application client. The TB bit does not affect the action taken for recovered data.

A read continuous (RC) bit of one indicates the device server shall transfer the entire requested length of data without adding delays to perform error recovery procedures. This implies that the device server may send data that is erroneous or fabricated in order to maintain a continuous flow of data. The device server shall assign priority to this bit over conflicting error control bits (EER, DCR, DTE, and PER) within this byte.

NOTE 33 Fabricated data may be data already in the buffer or any other device server-specific data. This bit is typically used in image processing, audio, or video applications.

A RC bit of zero indicates that error recovery operations that cause delays are acceptable during the data transfer. Data shall not be fabricated.

The individual bit definitions for EER, PER, DTE and DCR are contained in table 89. The combinations of these bits are explained in table 90.

Table 89 - Error recovery bit definitions

EER	PER	DTE	DCR	Description
1	-	-	-	An enable early recovery (EER) bit of one indicates that the device server shall use of the most expedient form of error recovery first. This bit only applies to data error recovery and it does not affect positioning retries and the message system error recovery procedures.
0	-	-	-	An EER bit of zero indicates that the device server shall use an error recovery procedure that minimizes the risk of mis-detection or mis-correction.
-	1	-	-	A post error (PER) bit of one indicates that the device server shall report recovered errors.
-	0	-	-	A PER bit of zero indicates that the device server shall not report recovered errors. Error recovery procedures shall be performed within the limits established by the error recovery parameters.
-	-	1	-	A disable transfer on error (DTE) bit of one indicates that the device server shall terminate the data phase upon detection of a recovered error.
-	-	0	-	A DTE bit of zero indicates that the device server shall not terminate the data phase upon detection of a recovered error.
-	-	-	1	A disable correction (DCR) bit of one indicates that error correction codes shall not be used for data error recovery.
-	-	-	0	A DCR bit of zero allows the use of error correction codes for data error recovery.

NOTE 34 An EER bit of one may imply an increase in the probability of mis-detection or mis-correction. An EER bit of zero typically means that the specified retry limit is exhausted prior to using error correction codes.

Table 90 - Combined error recovery parameter descriptions

EER	PER	DTE	DCR	Description
0	0	0	0	The full number of retries (specified in the read, write or verify retry count field) and error correction are attempted to recover the data (EER and DCR set to 0). A CHECK CONDITION is not reported at the completion of the command for recovered errors (PER set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurred, the data in the block with the unrecoverable error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only).
0	0	0	1	Error correction is disabled (DCR set to one) so only the full number of retries (specified in the read, write or verify retry count field) are attempted to recover the data (EER set to 0). A CHECK CONDITION is not reported at the completion of the command for recoverable errors (PER set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurred, the data in the block with the unrecoverable error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only).
0	0	1	0	Invalid mode (PER must be set to one if DTE is set to one). 1)
0	0	1	1	Invalid mode (PER must be set to one if DTE is set to one). 1)
0	1	0	0	The full number of retries (specified in the read, write or verify retry count field) and error correction are attempted to recover the data (EER and DCR set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurred, the data in the block with the unrecoverable error may or may not be transferred to the application client depending on the setting of the transfer block TB) bit(read operation only). A CHECK CONDITION with a sense key of RECOVERED ERROR is reported at the completion of the command for any recoverable error that occurs (PER set to 1). The information field in the sense data shall contain the logical block address of the last recovered error which occurred during the transfer.
0	1	0	1	Error correction is disabled (DCR set to one) so only the full number of retries (specified in the read, write or verify retry count field) are attempted to recover the data (EER set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurred, the data in the block with the unrecoverable error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only). A CHECK CONDITION with a sense key of RECOVERED ERROR is reported at the completion of the command for any recoverable error that occurs (PER set to 1). The information field in the sense data shall contain the logical block address of the last recovered error which occurred during the transfer.

Table 90 - Combined error recovery parameter descriptions (continued)

EER	PER	DTE	DCR	Description
0	1	1	0	The full number of retries (specified in the read, write or verify retry count field) and error correction are attempted to recover the data (EER and DCR set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted if any error (recoverable or unrecoverable) is detected (DTE set to 1). The information field in the sense data shall contain the logical block address of the block in error. If an unrecoverable data error occurs the data in the block with the error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only).
0	1	1	1	Error correction is disabled (DCR set to one) so only the full number of retries (specified in the read, write or verify retry count field) are attempted to recover the data (EER set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted if any error (recoverable or unrecoverable) is detected (DTE set to 1). The information field in the sense data shall contain the logical block address of the block in error. If an unrecoverable data error occurs the data in the block with the error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only).
1	0	0	0	The fewest possible retries and error correction are attempted to recover the data (EER set to one and DCR set to 0). A CHECK CONDITION is not reported at the completion of the command for recoverable errors (PER set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurred, the data in the block with the unrecoverable error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only).
1	0	0	1	Invalid mode (DCR must be set to zero if EER is set to one). 1)
1	0	1	0	Invalid mode (PER must be set to one if DTE is set to one). 1)
1	0	1	1	Invalid mode (PER must be set to one if DTE is set to one). 1)
1	1	0	0	The fewest possible retries and error correction are attempted to recover the data (EER set to one and DCR set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurred, the data in the block with the unrecoverable error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only). A CHECK CONDITION with a sense key of RECOVERED ERROR is reported at the completion of the command for any recoverable error that occurs (PER set to 1). The information field in the sense data shall contain the logical block address of the last recovered error which occurred during the transfer.
1	1	0	1	Invalid mode (DCR must be set to zero if EER is set to one). 1)

Table 90 - Combined error recovery parameter descriptions (concluded)

EER	PER	DTE	DCR	Description
1	1	1	0	The fewest possible retries and error correction are attempted to recover the data (EER set to one and DCR set to 0). The command terminates with CHECK CONDITION status before the transfer count is exhausted if any error (recoverable or unrecoverable) is detected (DTE set to 1). The information field in the sense data shall contain the logical block address of the block in error. If an unrecoverable data error occurs the data in the block with the error may or may not be transferred to the application client depending on the setting of the transfer block (TB) bit (read operation only).
1	1	1	1	Invalid mode (DCR must be set to zero if EER is set to one). 1)

1) If an invalid mode for the error recovery combination is sent by the application client the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The read and write retry count fields specify the number of times that the device server shall attempt its recovery algorithm during read and write operations, respectively. If the retry count field and the recovery time limit field are both specified in a MODE SELECT command, the field that requires the least time for data error recovery actions shall have priority.

The correction span field specifies the size, in bits, of the largest data error burst for which data error correction may be attempted. A correction span of zero specifies that the device server shall use its default value or that this field is not supported.

The head offset count field specifies in two's-complement notation an incremental offset position from the track center to which the heads shall be moved. The effect of this field on write operations is unspecified. A head offset count of zero indicates that no offset is specified. A positive value indicates moving in the direction of increasing logical block addresses. A negative value indicates moving in the direction of decreasing logical block addresses. Any value specified in this field does not preclude the device server from using positive or negative head offset during error recovery. However, after any error recovery is completed the device server shall return the head offset to the value specified in this field.

> **NOTE 35** The degree of offset for each incremental value and the number of valid values are logical unit specific. It is recommended that the number of valid values be equal for the positive and negative head offset counts.

The device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST if an unsupported head offset value is specified. The valid bit shall be set to one and the information field shall be set to the positive value of the maximum head offset count that is supported. The device server shall set the valid bit to zero if it cannot determine the maximum head offset count supported.

> **NOTE 36** If the device server does not support this field, it returns a zero value in the MODE SENSE command.)

The data strobe offset count field specifies in two's-complement notation an incremental position to which the recovered data strobe shall be adjusted from its nominal setting. The effect of this field on write operations is unspecified. A value of zero indicates that no data strobe offset is specified. A positive value indicates movement in a positive direction as defined by the device server. A negative value indicates movement in the negative direction as defined by the device server. Any value specified in this field does not preclude the device server from using positive or negative data strobe offset during

error recovery. However, after any error recovery is completed the device server shall return the data strobe offset to the value specified in this field.

NOTE 37 The degree of offset for each incremental value and the number of valid values are logical unit specific. It is recommended that the number of valid values be equal for the positive and negative data strobe offset counts.

The device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST if an unsupported data strobe offset count value is specified. The valid bit shall be set to one and the information field shall be set to the positive value of the maximum data strobe offset count that is supported. The device server shall set the valid bit to zero if it cannot determine the maximum data strobe offset supported.

NOTE 38 If the device server does not support the data strobe offset count field, it returns a zero value in the MODE SENSE command.

The recovery time limit field specifies in increments of one ms the maximum time duration that the device server shall use for data error recovery procedures. The device server may round this value as described in 7.5.4. The limits in this field specifies the maximum error recovery time allowed for any individual logical block. A recovery time limit of zero specifies that the device server shall use its default value.

If both retry count and recovery time limit are specified, the field that specifies the recovery action of least duration shall have priority.

7.1.3.7 Rigid disk drive geometry page

The rigid disk drive geometry page (see table 91) specifies parameters for direct-access logical units employing a rigid disk drive.

Table 91 - Rigid disk drive geometry page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved		Page code (04h)				
1				Page length (16h)				
2	(MSB)							
3				Number of cylinders				
4								(LSB)
5				Number of heads				
6	(MSB)							
7				Starting cylinder-write precompensation				
8								(LSB)
9	(MSB)							
10				Starting cylinder-reduced write current				
11								(LSB)
12	(MSB)							
13				Drive step rate				
14								(LSB)
15	(MSB)							
16				Landing zone cylinder				
17								(LSB)
18				Reserved			RPL	
19				Rotational offset				
20				Reserved				
21	(MSB)							
22				Medium rotation rate				
23								(LSB)
				Reserved				
				Reserved				

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

> **NOTE 39** This page is intended to define drive geometry parameters of rigid disk drives. It may be used for other logical units if applicable.

The number of cylinders field defines the number of physical cylinders used for data storage.

The number of heads field defines the physical number of heads used for data storage. Heads used exclusively for servo information are excluded.

The starting cylinder for write precompensation field is the physical cylinder at which write precompensation is to begin. The first cylinder is number zero. If the starting cylinder for write precompensation is equal to the value in the number of cylinders field, write precompensation shall be disabled by the device server.

The starting cylinder for reduced write current field is the physical cylinder at which write current is reduced. The first cylinder is number zero. If the starting cylinder for reduced write current is equal to the value in the number of cylinders field, reduced write current shall be disabled by the device server.

The drive step rate field indicates the step rate in 100 ns increments. The device server shall use the drive step rate, greater than or equal to the drive step rate specified. If the device server rounds this field it shall terminate the command as described in 7.5.4. A value of zero requests the device server to set its default value.

The landing zone cylinder field indicates two's complement location where the device server shall position the disk heads. A negative value indicates that the heads are positioned below cylinder zero by that number of cylinders. A positive value greater than the number of cylinders indicates that the heads are positioned beyond the cylinders used for data storage at the cylinder location specified. A zero indicates that the default should be used.

The rotational position locking (RPL) field is used for spindle synchronization as defined in table 92. See K of the ANSI X3.170A- 1991 for further information on synchronized spindles.

Table 92 - Rotational position locking

RPL	Description
00b	Indicates that spindle synchronization is disabled or not supported
01b	The device server operates as a synchronized-spindle slave
10b	The device server operates as a synchronized-spindle master
11b	The device server operates as a synchronized-spindle master control

NOTE 40 The signals and connectors used for rotational position locking are external to the SCSI bus and are not part of this International Standard.

If a device server fails to achieve synchronization it shall create a unit attention condition to all application clients. The sense key shall be set to UNIT ATTENTION and the additional sense code set to RPL STATUS CHANGE.

If subsequent to achieving synchronization the device server detects a change of synchronization:

a) and, if the logical unit is not presently executing an I/O process for the application client, the device server shall create a unit attention condition. The sense key shall be set to UNIT ATTENTION and the additional sense code set to RPL STATUS CHANGE.

b) and, if the logical unit is presently executing an I/O process and no other error occurs, then the device server shall return CHECK CONDITION status. The sense key shall be set to RECOVERED ERROR if the device server is able to complete the I/O process or HARDWARE ERROR if the device server is unable to complete the I/O process. The additional sense code is set to RPL STATUS CHANGE.

The rotational offset indicates the amount of rotational skew that the device server shall use when synchronized. The rotational skew is applied in the retarded direction (lagging the synchronized spindle master control). The value in the field is the numerator of a fractional multiplier that has 256 as its denominator (e.g. a value of 128 indicates a one-half revolution skew). A value of zero indicates that rotational offset shall not be used. This value may be rounded as defined in 7.5.4. The rotational offset is not used when a device server is configured as synchronized-spindle master.

The medium rotation rate indicates the speed at which the medium rotates. The unit of measure is rotations per minute (e.g. 3 600 rpm).

7.1.3.8 Verify error recovery page

The verify error recovery page (see table 93) specifies the error recovery parameters the device server shall use during the VERIFY command, the verify operation of the WRITE AND VERIFY command and the verify operation of the COPY AND VERIFY command.

Table 93 - Verify error recovery page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (07h)					
1	Parameter length (0Ah)							
2	Reserved				EER	PER	DTE	DCR
3	Verify retry count							
4	Verify correction span							
5	Reserved							
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	(MSB) Verify recovery time limit							
11								(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 device server is capable of saving the page in a non-volatile vendor-specific location.

The AWRE bit as defined in the read-write error recovery page (see 7.1.3.6) applies to the WRITE AND VERIFY command. The VERIFY and COMPARE commands shall not perform automatic reallocation. The COPY and COPY AND VERIFY commands might or might not perform automatic reallocation depending on the settings of the mode parameters.

The EER, PER, DTE and DCR bits are defined in 7.1.3.6. The combinations of these bits is defined in 93.3.6.

The verify retry count field specifies the number of times that the device server shall attempt its recovery algorithm during a verify operation. If the verify retry count and the verify recovery time limit are both specified, the one that requires the least time for data error recovery actions shall have priority.

The verify correction span field specifies the size, in bits, of the largest burst data error for which data error correction may be attempted. If the device server does not implement this field, a value of zero is returned in MODE SENSE data.

The verify recovery time limit field specifies in increments of one millisecond the maximum time duration that the device server shall use error recovery procedures to recover data for an individual logical block. The device server may round this value as described in 7.5.4. If the verify retry count and the verify recovery time limit are both specified, the one that requires the least time for data error recovery actions shall have priority.

NOTE 41 To disable all types of correction and retries the application client should set the EER bit to zero, the PER, DTE and DCR bits to one and the number of retries and recovery time limit to zero.

7.2 Parameters for optical memory logical units

7.2.1 Diagnostic parameters

This subclause defines the descriptors and pages for diagnostic parameters used with optical memory logical units.

The diagnostic page codes for optical memory logical units are defined in table 94.

Table 94 - Diagnostic page codes

Page code	Description	Subclause
00h	Supported diagnostic pages	SPC
40h	Translate address page - SEND DIAGNOSTIC	7.1.1.1
40h	Translate address page - RECEIVE DIAGNOSTIC	7.1.1.2
41h	Drive status page - SEND DIAGNOSTIC	7.1.1.3
41h	Drive status page - RECEIVE DIAGNOSTIC	7.1.1.4
01h - 3Fh	Reserved (for all device type pages)	
42h - 7Fh	Reserved	
80h - FFh	Vendor-specific pages	

7.2.2 Log parameters

This subclause defines the descriptors and pages for log parameters used with optical memory logical units.

The log page codes for optical memory logical units are defined in table 95.

Table 95 - Log page codes

Page code	Description	Subclause
01h	Buffer over-run/under-run page	SPC
03h	Error counter page (read) page	SPC
04h	Error counter page (read reverse) page	SPC
05h	Error counter page (verify) page	SPC
02h	Error counter page (write) page	SPC
07h	Last n error events page	SPC
06h	Non-medium error page	SPC
00h	Supported log pages	SPC
08h	Format status page	7.1.2.1
09h - 2Fh	Reserved	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

7.2.3 Mode parameters

This subclause defines the descriptors and pages for mode parameters used with optical memory logical units.

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

The medium-type code field is contained in the mode parameter header (see the SPC). Table 96 defines the medium-type code values used for optical memory logical units.

Table 96 - Optical memory medium-type codes

Code	Description
00h	Default (only one medium type supported)
01h	Optical read-only medium
02h	Optical write-once medium
03h	Optical reversible or erasable medium
04h	Combination of read-only and write-once medium
05h	Combination of read-only and reversible or erasable medium
06h	Combination of write-once and reversible or erasable medium
07h - 7Fh	RESERVED
80h - FFh	Vendor-specific

The device specific parameter field is contained in the mode parameter header (see the SPC). Table 97 defines the device specific parameter values used for optical memory logical units.

Table 97 - Optical memory logical unit specific parameter

Bit	7	6	5	4	3	2	1	0
	WP	Reserved		DPOFUA	Reserved			EBC

When used with the MODE SELECT command the WP bit is not defined.

When used with the MODE SENSE command, a write protected (WP) bit of zero indicates that the medium is write enabled. A WP bit of one indicates that the medium is write protected. For read-only media the WP bit is reserved.

When used with the MODE SELECT command the DPOFUA bit is reserved.

When used with the MODE SENSE command, a DPOFUA bit of one indicates that the device server supports the DPO and FUA bits (see 6.1.6).

For the MODE SELECT command, an enable blank check (EBC) bit of zero advises the device server to disable the blank checking operation of the medium during write operations or during an UPDATE BLOCK command. An EBC bit of one enables blank checking. If a non-blank block is found during a write operation, the command shall be terminated with a CHECK CONDITION status and the sense key shall be set to BLANK CHECK. If a blank block is found during an UPDATE BLOCK command, the command shall be terminated with a CHECK CONDITION status, and the sense key shall be set to BLANK CHECK. For read-only media, the EBC bit is reserved.

For the MODE SENSE command, an enable blank check (EBC) bit of zero indicates that blank checking of the medium during write operations is disabled. An EBC bit of one indicates that blank checking during write and update operations is enabled. For read-only media, the EBC bit is reserved.

The density code field is contained in the mode parameter block descriptor (see the SPC). Table 98 defines the density code values used for optical memory logical units.

Table 98 - Optical memory density codes

Density code	Optical media							
00h	Default density (currently mounted density)							
	Diameter mm (in)	Type	Sector size	Tracks	Sides	Servo	Reference standard	Notes
01h	86 (3,5)	R/W	512/1 024	12 500	1		660-D	1
02h	89 (3,5)	R/W	512/1 024	12 500	2		581-D	1
03h	130 (5,25)	R/W	512/1 024	18 750	2	CS	607-I	1
04h	130 (5,25)	W-O	512/1 024	30 000	2	SS	655-D	1,2
05h	130 (5,25)	W-O	512/1 024	20 000	2	SS	659-D	1,3
06h	130 (5,25)	W-O	512/1 024	18 750	2	CS	457-D	1
07h	200 (8,0)						407-D	1,4
08h	300 (12,0)		1 024		2		408-D	1,4
09h	356 (14,0)		1 024	56 350	2		456-D	1
Codes 80h - FFh are vendor-specific, all other codes are reserved								

Key:	Type	Description	Servo	Description
	R/W	Erasable	CS	Continuous servo
	W-O	Write once	SS	Sampled servo
	R/O	Read only		

NOTES
1 The reference standard refers to an ANSI X3B11 project under development. The project may have several documents describing different aspects of the media. Please contact the Secretariat for current information on status and availability.
2 RZ modulation.
3 4/15 modulation.
4 This project is not presently active.

The mode page codes for optical memory logical units are shown in table 99.

Table 99 - Mode page codes

Page code	Description	Subclause
08h	Caching page	7.1.3.1
0Ah	Control mode page	SPC
02h	Disconnect-reconnect page	SPC
1Ch	Informational exceptions control page	SPC
0Bh	Medium types supported page	7.1.3.4
06h	Optical memory page	7.2.3.1
09h	Peripheral device page	SPC
0Dh	Power condition page	7.1.3.6
01h	Read-write error recovery page	7.1.3.6
07h	Verify error recovery page	7.1.3.8
03h - 05h	Reserved	
0Ch - 1Bh	Reserved	
1Dh - 1Fh	Reserved	
00h	Vendor-specific (does not require page format)	
20h - 3Eh	Vendor-specific (page format required)	
3Fh	Return all pages (valid only for the MODE SENSE command)	

7.2.3.1 Optical memory page

The optical memory page (see table 100) defines parameters for control of optical memory logical units.

Table 100 - Optical memory page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (06h)					
1	Parameter length (02h)							
2	Reserved							RUBR
3	Reserved							

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

A report updated block read (RUBR) bit of zero indicates the device server shall not report an error when a command performs a successful read of a block that has been updated. A RUBR bit of one indicates the device server shall terminate a command that performs a read of a block that has been updated with CHECK CONDITION status, the sense key set to RECOVERED ERROR, the additional sense code set to UPDATED BLOCK READ, and the data shall be transferred to the application client. The default state of the RUBR bit for write-once logical units (as reported in the INQUIRY command) shall be one.

7.3 Parameters for write-once logical units

Refer to the parameters for optical memory logical units (see 7.2).

•