draft proposed **T10**
American National Standard **995D**
for information systems - **Revision 11a**

SCSI-3 Primary Commands

28 March 1997

Secretariat: National Committee for Information Technology Standards (NCITS)

Notice:

This is a draft proposed American National Standard of Accredited Standards Committee NCITS. As such, this is not a completed standard. The T10 Technical Committee may modify this document as a result of comments received during public review and its approval as a standard.

Project Editor:

Ralph O. Weber
Symbios Logic
17304 Preston Rd., Suite 635
Dallas, TX 75252
USA
Telephone: 972-733-3594
Facsimile: 972-713-8121
Email: ROWeber@ACM.org

Reference number
ISO/IEC **** : 199x
ANSI X3.301 - 199x
Printed March, 29, 1997 10:06am

POINTS OF CONTACT:

T10 Chair
John B. Lohmeyer
Symbios Logic Inc.
4420 ArrowsWest Drive
Colo Spgs, CO 80907-3444

Tel: (719) 533-7560
Fax: (719) 533-7036
Email: john.lohmeyer@symbios.com

T10 Vice-Chair
Lawrence J. Lamers
Adaptec
691 South Milpitas Blvd.
Milpitas, CA 95035

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

NCITS Secretariat
NCITS Secretariat
1250 Eye Street, NW Suite 200
Washington, DC 20005

Telephone: 202-737-8888
Facsimile: 202-638-4922
Email: x3sec@itic.nw.dc.us

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

SCSI Bulletin Board
719-533-7950

T10 WWW page
http://www.symbios.com/x3t10 or http://www.symbios.com/t10

T10 ftp
ftp.symbios.com
in the /pub/standards/io/x3t10/drafts/* directories

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

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

ANSI (r)
X3.301 - 199x

American National Standard
for Information Systems-

SCSI-3 Primary Commands (SPC)

Secretariat
Information Technology Industry Council

Approved mm dd yy

American National Standards Institute, Inc.

Abstract

This standard defines the device model for all SCSI devices. This standard defines the SCSI commands that are basic to every device model and the SCSI commands that may apply to any device model.

The processor device model is defined in this standard. Some target SCSI devices may require a host implementation of the processor device model to support the Asynchronous Event Reporting capability defined in the SCSI-3 Architecture Model.

American National Standard

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

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

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

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

CAUTION: The developers of this standard have requested that holder's of patents that may be required for the implementation of this 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 have been issued 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 not required to avoid infringement in the use of this standard.

Published by
American National Standards Institute
11 West 42nd Street, New York, NY 10036

Contents

Annexes

Tables

Figures

Foreword

This foreword is not part of American National Standard X3.301 - 199x.

The SCSI command set is designed to provide efficient peer-to-peer operation of SCSI devices (disks, tapes, printers, etc.) by an operating system. The SCSI command set assumes an underlying command-response protocol.

The SCSI command set provides multiple operating systems concurrent control over one or more SCSI devices. However, proper coordination of activities between the multiple operating systems is critical to avoid data corruption. Commands that assist with coordination between multiple operating systems are described in this standard. However, details of the coordination are beyond the scope of the SCSI command set.

This standard defines the device model for all SCSI devices. This standard defines the SCSI commands that are basic to every device model and the SCSI commands that may apply to any device model.

The processor device model is defined in this standard. Some target SCSI devices may require a host implementation of the processor device model to support the Asynchronous Event Reporting capability defined in the SCSI-3 Architecture Model. Thus, the SCSI processor device commands are defined in this standard.

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

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, X3.301 - 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 NCITS 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

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the NCITS Secretariat, National Committee for Information Technology Standards, 1250 Eye Street, NW, Suite 200, Washington, DC 20005-3922.

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

James D. Converse, Chair
Donald C. Loughry, Vice-Chair
Joanne M. Flanagan, Secretary

Organization Represented	*Name of Representative*
American Nuclear Society	Geraldine C. Main
	Sally Hartzell (Alt.)
AMP, Inc	Edward Kelly
	Charles Brill (Alt.)

Apple Computer	Karen Higginbottom
Association of the Institute for Certification of Professionals (AICCP)	Kenneth Zemrowski
AT&T/NCR	Thomas W. Kern
	Thomas F. Frost (Alt.)
Boeing Company	Catherine Howells
	Andrea Vanosdoll (Alt.)
Bull HN Information Systems, Inc.	William George
Compaq Computer Corporation	James Barnes
Digital Equipment Corporation	Delbert Shoemaker
	Kevin Lewis (Alt.)
Eastman Kodak	James D. Converse
	Michael Nier (Alt.)
GUIDE International	Frank Kirshenbaum
	Harold Kuneke (Alt.)
Hewlett-Packard	Donald C. Loughry
Hitachi America, Ltd.	John Neumann
	Kei Yamashita (Alt.)
Hughes Aircraft Company	Harold L. Zebrack
IBM Corporation	Joel Urman
	Mary Anne Lawler (Alt.)
National Communication Systems	Dennis Bodson
National Institute of Standards and Technology	Robert E. Roundtree
	Michael Hogan (Alt.)
Northern Telecom, Inc.	Mel Woinsky
	Subhash Patel (Alt.)
Neville & Associates	Carlton Neville
Recognition Technology Users Association	Herbert P. Schantz
	G. Edwin Hale (Alt.)
Share, Inc.	Gary Ainsworth
	David Thewlis (Alt.)
Sony Corporation	Michael Deese
Storage Technology Corporation	Joseph S. Zajaczkowski
	Samuel D. Cheatham (Alt.)
Sun Microsystems	Scott Jameson
	Gary Robinson (Alt.)
3M Company	Eddie T. Morioka
	Paul D. Jahnke (Alt.)
Unisys Corporation	John L. Hill
	Stephen P. Oksala (Alt.)
U.S. Department of Defense	William C. Rinehuls
	C. J. Pasquariello (Alt.)
U.S. Department of Energy	Alton Cox
	Lawrence A. Wasson (Alt.)
U.S. General Services Administration	Douglas Arai
	Larry L. Jackson (Alt.)
Wintergreen Information Services	Joun Wheeler
Xerox Corporation	Dwight McBain
	Roy Peirce (Alt.)

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

John B. Lohmeyer, Chair
Lawrence J. Lamers, Vice-Chair
Ralph O. Weber, Secretary

Norm Harris
Edward Fong
Charles Brill
Michael Wingard
Jan V. Dedek
Dennis Pak
Clifford E. Strang Jr.
Gerry Johnsen
Ian Morrell
Joe Chen
Edward Haske
Peter Johansson
Michael Bryan
Louis Grantham
Charles Monia
Robert Reisch
Edward Lappin
Gary R. Stephens
Robert Liu
Stephen Holmstead
S. Nadershahi
David McFadden
George Penokie
Duncan Penman
Geoffrey Barton
Dean Wallace
Robert Bellino
Pete McLean
Steve D. Schueler
Joe Dambach
Chris D'Iorio
Dennis Van Dalsen
Edward A. Gardner
Stephen F. Heil
Skip Jones
James McGrath
Gene Milligan
Stephen G. Finch
Erich Oetting
Robert N. Snively
Patrick Mercer
John Moy Tokuyuki
Totani Gary
M. Watson
Kenneth J. Hallam
Paul D. Aloisi
Jeff Stai

Doug Piper
Mike Eneboe (Alt.)
Dhiru N. Desai (Alt.)
Nicos Syrimis (Alt.)
Timothy Feldman (Alt.)
Marc A. Noblitt (Alt.)
William Dallas (Alt.)
Douglas Hagerman (Alt.)
Dr. William Ham (Alt.)
Mike Chennery (Alt.)
Tim Norman (Alt.)
Kevin R. Pokorney (Alt.)
Paul Boulay (Alt.)
Ms. Nancy Cheng (Alt.)
Zane Daggett (Alt.)
Anthony Yang (Alt.)
Ruben Yomtoubian (Alt.)
Dan Colegrove (Alt.)
Frank Samela (Alt.)
Wayne Baldwin (Alt.)
Jack Chen (Alt.)
Brian N. Davis (Alt.)
Brian Johnson (Alt.)
John Masiewicz (Alt.)
Tom Hanan (Alt.)
Jeffrey L. Williams (Alt.)
Devon Worrell (Alt.)
Richard Kalish (Alt.)
Peter Haas (Alt.)
Bob Whiteman (Alt.)
Bill Mable (Alt.)
Gary Porter (Alt.)
Ron Roberts (Alt.)
Jaff Lin (Alt.)
Raymond Gilson (Alt.)
Dennis Lang (Alt.)
John Geldman (Alt.)
Roger Wang (Alt.)
Joseph Wach (Alt.)
Michael Smith (Alt.)
Rick Bohn (Alt.)
I. Dal Allan (Alt.)
David Andreatta (Alt.)
Andy Chen (Alt.)
J. R. Sims (Alt.)
Danny Yeung (Alt.)
Thomas J. Kulesza (Alt.)

John P. Scheible (Alt.)
Pat LaVarre (Alt.)
Shufan Chan (Alt.)
Chuck Grant (Alt.)
John Cannon (Alt.)
Jay Neer (Alt.)
Mike Hetzel (Alt.)
Dr. Akira James Miura (Alt.)
Ting Li Chan (Alt.)
Mark Evans (Alt.)
Gerald Houlder (Alt.)
Dave Guss (Alt.)
Vit Novak (Alt.)
Greg Kapraun (Alt.)
Akram Atallah (Alt.)
Pete Tobias (Alt.)
Arlan P. Stone (Alt.)
Matthew Thomas (Alt.)
Tak Asami (Alt.)
E.J. Mondor (Alt.)

Introduction

The SCSI-3 Primary Commands (SPC) standard is divided into ten clauses:

Clause 1 is the scope.

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

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

Clause 4 describes the conceptual relationship between this document and the SCSI-3 Architecture Model.

Clause 5 describes the command model for all SCSI devices.

Clause 6 describes the command model for processor type SCSI devices.

Clause 7 defines the commands that may be implemented by any SCSI device.

Clause 8 defines the parameter data formats that may be implemented by any SCSI device.

Clause 9 defines the commands that may be implemented by a processor type SCSI device.

Clause 10 defines the parameter data formats that may be implemented by a processor type SCSI device.

The annexes provide information to assist with implementation of the SCSI-3 Primary Commands standard. The information in the annexes applies to all the SCSI-3 command standards. See 3.1.11 for more information about other SCSI-3 command standards.

American National Standard X3.301 - 199x

American National Standard for Information Systems -
Information Technology -

SCSI- 3 Primary Commands (SPC)

1 Scope

The SCSI-3 family of standards provides for many different types of SCSI devices (disks, tapes, printers, scanners, and many more). This standard defines a device model that is applicable to all SCSI devices. Other SCSI-3 command standards (see 3.1.11) expand on the general SCSI device model in ways appropriate to specific types of SCSI devices.

This standard defines the SCSI commands that are mandatory and optional for all SCSI devices. This standard also defines the SCSI commands that may apply to any device model.

Since a host processor is a part of any SCSI domain, the processor device model is defined in this standard. The commands that may be implemented by an SCSI processor device likewise are defined in this standard. Some target SCSI devices may require a host implementation of the processor device model to support the Asynchronous Event Reporting capability defined in the SCSI-3 Architecture Model.

Figure 1 shows the relationship of this standard to the other standards and related projects in the SCSI-3 family standards as of the publication of this standard.

The roadmap in figure 1 is intended to show the general applicability of the documents to one another.

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

The term SCSI-3 refers collectively to the following documents that fall under the jurisdiction of T10:

- SCSI-3 Architecture Model	SAM	[T10/994-D]
- SCSI-3 Block Commands	SBC	[T10/996-D]
- SCSI-3 Stream Commands	SSC	[T10/997-D]
- SCSI-3 Graphics Commands	SGC	[T10/998-D]
- SCSI-3 Medium Changer Commands	SMC	[T10/999-D]
- SCSI-3 Controller Commands	SCC	[T10/1047-D]
- SCSI-3 Multimedia Commands	MMC	[T10/1048-D]
- SCSI-3 Enclosure Services	SES	[T10/1212-D]
- SCSI-3 Primary Commands	SPC	[T10/995-D]
- SCSI-3 Parallel Interface	SPI	[T10/855-D]
- SCSI-3 Fast-20 Parallel Interface		[T10/1071-D]
- SCSI-3 Interlocked Protocol	SIP	[T10/856-D]
- SCSI-3 Serial Bus Protocol	SBP	[T10/992-D]
- SCSI-3 Fiber Channel Protocol	FCP	[T10/993-D]

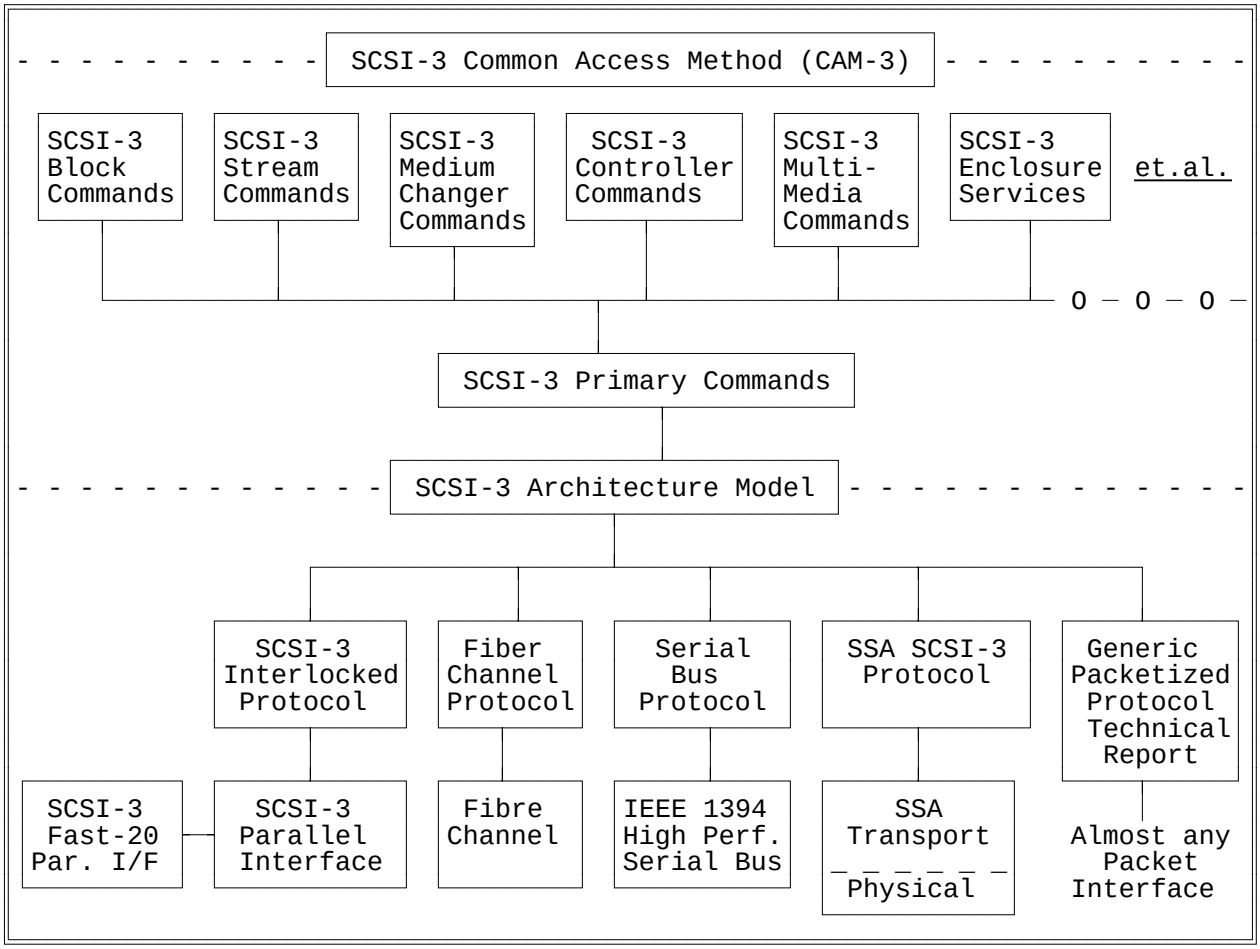

Figure 1 – SCSI-3 document roadmap

2 Normative references

2.1 Document and draft document availability information

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

At the time of publication, NCITS practice was to make working draft standards and draft proposed American National Standards available through Global Engineering at 800-854-7179 (toll free phone), 303-792-2181 (phone) or 303-792-2192 (fax).

2.2 Normative approved references for mandatory features

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

- SCSI-3 Architecture Model	SAM	X3.270 - 199x

2.3 Normative approved references for optional features

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

- SCSI-3 Architecture Model	SAM	X3.270 - 199x
- SCSI-2 Small Computer System Interface	SCSI-2	X3.131 - 1994
- Fibre Channel Physical	FC-PH	X3.230 - 1994
- Extended Unique Identifier, 64-bit	EUI-64	IEEE ????

2.4 Normative references under development for optional features

At the time of publication, the following standards were still under development and expected to contain provisions which, through reference in the text, constitute optional provisions of this standard. For information on the current status of the document, or regarding availability, contact the NCITS Secretariat at 202-626-5738 (phone), 202-638-4922 (fax) or email x3sec@itc.nw.dc.us.

- SCSI-3 Medium Changer Commands	SMC	[T10/999-D]

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

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

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

3.1.3 attached medium changer: A medium changer that is attached to and accessed through some other type of SCSI device. (See 5.5.)

3.1.4 asynchronous event reporting: Asynchronous event reporting is used by a logical unit to signal an initiator that an asynchronous event has occurred. The mechanism by which asynchronous event reporting works is protocol-specific. A detailed definition of AER may be found in SAM.

3.1.5 auto contingent allegiance: The condition of a task set following the return of a CHECK CONDITION or COMMAND TERMINATED status. A detailed definition of ACA may be found in SAM.

3.1.6 autosense data: The sense data that is automatically delivered to the application client by the device server in a protocol-specific manner when a command completes with a CHECK CONDITION or COMMAND TERMINATED status (see 4.1 and SAM).

3.1.7 blocked task: A blocked task is a task that is in the blocked state, as defined in SAM. Tasks become blocked when an auto contingent allegiance condition occurs. The blocked state ends when the ACA condition is cleared. A detailed definition of the blocked task state may be found in SAM.

3.1.8 byte: Indicates an 8-bit construct.

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

3.1.10 command descriptor block: The structure up to 16 bytes in length used to communicate commands from an application client to a device server.

3.1.11 command standard: An SCSI-3 standard that defines another device type models, commands, and parameter data; e.g., SBC, SCC, SGC, SMC, SSC, MMC, SES, etc. (see clause 1).

3.1.12 copy manager: The device server that receives a COPY, COMPARE or COPY AND VERIFY command and performs the operation thus requested.

3.1.13 data-in buffer: The buffer identified by the application client to receive data from the device server during the execution of a command (see 4.1 and SAM).

3.1.14 data-out buffer: The buffer identified by the application client to supply data that is sent from the application client to the device server during the execution of a command (see 4.1 and SAM).

3.1.15 data packet: The data transferred in the Data-In Buffer associated with a processor device RECEIVE command, or during the Data-Out Buffer associated with a processor device SEND command. A data packet often contains information at the beginning or end of the packet that describes the contents of the packet. A data packet may contain control or status information for the destination device.

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

3.1.17 device service request: A request, submitted by an application client, conveying an SCSI command to a device server. A detailed definition of a device service request may be found in SAM.

3.1.18 device service response: The response returned to an application client by a device server on completion of an SCSI command. A detailed definition of a device service response may be found in SAM.

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

3.1.20 element: An addressable physical component of a medium changer device that can serve as the location of a removable unit of data storage medium. A detailed definition of an element may be found in SMC.

3.1.21 extent: An extent is a specified number of logical blocks, typically identified by a starting logical block address and a count of the number of blocks in the extent.

3.1.22 enabled task state: The enabled task state is the only task state in which a task may make effective progress towards completion. A detailed definition of the enabled task state may be found in SAM.

3.1.23 field: A group of one or more contiguous bits.

3.1.24 hard reset: A target response to a reset event or TARGET RESET task management function. A detailed definition of hard reset may be found in SAM.

3.1.25 host: A device with the characteristics of a primary computing device, typically a personal computer, workstation, minicomputer, mainframe computer, or auxiliary computing device or server. Although there are a few exceptions, a host typically functions as an initiator.

3.1.26 idle condition: When 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.

3.1.27 initiator: An SCSI device containing application clients that originate device service requests to be processed in a device server. A detailed definition of an initiator may be found in SAM.

3.1.28 initiator role agent: A component of the service delivery subsystem that carries out the actions of a request following the initiator rules of the protocol.

3.1.29 linked command: One in a series of SCSI commands executed by a single task, which collectively make up a discrete I/O operation. A detailed definition of a linked command may be found in SAM.

3.1.30 logical unit: An externally addressable entity within a target that implements an SCSI device model and contains a device server. A detailed definition of a logical unit may be found in SAM.

3.1.31 logical unit identifier: An object that is part of the SAM definition of a logical unit. A logical unit identifier uniquely identifies a logical unit in an SCSI domain. Detailed definitions of SCSI domain and logical unit identifier may be found in SAM.

3.1.32 logical unit number: An encoded 64-bit identifier for a logical unit. A detailed definition of a logical unit number may be found in SAM.

3.1.33 medium: Except where noted, the usage of medium in this standard is synonymous with media information, as defined by SAM; i.e., information stored within an SCSI device, which is non-volatile (retained through a power cycle) and accessible to an initiator through the execution of SCSI commands.

3.1.34 medium changer: A medium changer mechanizes the movement of media to and from the device that records on or reads from the media. A detailed definition of a medium changer may be found in SMC.

3.1.35 one: The logical true condition of a variable.

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

3.1.37 protocol-specific: Requirements for the referenced item are defined by an SCSI-3 protocol standard. A detailed definition of protocol-specific may be found in SAM.

3.1.38 resource: A part of a processor device required to operate on or store a data packet.

3.1.39 SCSI device: A device that is connected to a service delivery subsystem and supports an SCSI application protocol. A detailed definition of an SCSI device may be found in SAM.

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

3.1.41 sense data: Data describing an error or exceptional device condition that a device server delivers to an application client (see 7.20). Sense data may be delivered in response to a REQUEST SENSE command or as Autosense Data.

3.1.42 service action: A request describing a unit of work to be performed by a device server. A service action is an extension of a command. See SAM for a detailed definition of a command.

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

3.1.44 standby condition: When a logical unit is capable of accepting commands, but media is not immediately accessible (e.g., spindle is stopped).

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

3.1.46 system: A system is one or more SCSI domains operating as a single configuration.

3.1.47 target: An SCSI device containing logical units that receive and execute commands from an initiator. A detailed definition of a target may be found in SAM.

3.1.48 target role agent: A component of the service delivery subsystem that carries out the actions of a request following the target rules of the protocol.

3.1.49 task: An object within a logical unit that represents the work associated with a command or a group of linked commands. A detailed definition of a task may be found in SAM.

3.1.50 task set: A group of tasks within a logical unit, whose interaction is dependent on the queuing and ACA rules defined in SAM.

3.1.51 third-party: When used in reference to COPY commands, third-party means a COPY command issued to one device to perform a copy operation between two other devices. When used in reference to RESERVE, or RELEASE commands, third-party means a reservation made on behalf of another device (e.g., a processor device requests that a direct-access device reserve itself for use by a sequential-access device).

3.1.52 unit attention condition: A state that a logical unit maintains while it has asynchronous status information to report to one or more initiators. A detailed definition of the unit attention condition may be found in SAM.

3.1.53 vendor-specific: Something (e.g., a bit, field, code value, etc.) that is not defined by this standard and may be vendor defined.

3.1.54 zero: The logical false condition of a variable.

3.2 Acronyms

ACA Auto Contingent Allegiance (see 3.1.5)
AER Asynchronous Event Reporting (see 3.1.4)
ASC Additional Sense Code (see 7.20)
ASCQ Additional Sense Code Qualifier (see 7.20)
CDB Command Descriptor Block (see 3.1.10)
LSB Least significant bit
LUN Logical Unit Number (see 3.1.32)
MMC SCSI-3 Multi-Media Commands (see clause 1)
MSB Most significant bit
RAID Redundant Array of Independent Disks
SAM SCSI-3 Architecture Model (see clause 1)
SBC SCSI-3 Block Commands (see clause 1)
SCC SCSI-3 Controller Commands (see clause 1)
SCSI Either SCSI-2 or SCSI-3.
SCSI-2 The architecture defined by the Small Computer System Interface - 2 standard (ANSI X3.131 - 1994)
SCSI-3 The architecture defined by the family of standards described in clause 1
SES SCSI-3 Enclosure Services (see clause 1)
SGC SCSI-3 Graphic Commands (see clause 1)
SMC SCSI-3 Medium Changer Commands (see clause 1)
SPC SCSI-3 Primary Commands (this standard, see clause 1)
SSC SCSI-3 Stream Commands (see clause 1)
VPD Vital Product Data (see 8.4)
VS Vendor-Specific (see 3.1.53)

3.3 Keywords

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

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

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

3.3.4 may: A keyword that indicated flexibility of choice with no implied preference.

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

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

3.3.7 reserved: A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients may check reserved bits, bytes, words or fields for zero values and report errors if non-zero values are received. Receipt of reserved code values in defined fields shall be reported as error.

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

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

3.4 Conventions

Certain words and terms used in this standard have a specific meaning beyond the normal English meaning. These words and terms are defined either in 3.1 or in the text where they first appear. Names of commands, statuses, sense keys, additional sense codes, and additional sense code qualifiers are in all uppercase (e.g., REQUEST SENSE). Lower case is used for words having the normal English meaning.

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

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

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

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

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

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

4 General

This standard defines behaviors that are common to all SCSI device models (see clause 5). This standard defines the SCSI commands that are basic to more than one device model and the SCSI commands that may apply to any device model (see clause 7). This standard defines the parameters that are basic to more than one device model (see clause 8).

The processor device model (see clause 6), commands (see clause 9), and parameters (see clause 10) are defined in this standard.

4.1 The request-response model

The SCSI command set assumes an underlying request-response protocol. The fundamental properties of the request-response protocol are defined in the SCSI-3 Architecture Model (SAM). 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. As per SAM, the request-response protocol may be modelled as a procedure call, specifically:

Service response = Execute Command (Task Identifier, CDB, [Data-Out Buffer], Task Attributes, || [Data-In Buffer], [Autosense Data], [Autosense Return Flag], Status)

SAM defines all of the inputs and outputs in the procedure call above. As they may apply to any SCSI device, this standard defines the contents of the following procedure inputs and outputs; CDB, Data-Out Buffer, Data-In Buffer, and Autosense Data. This standard **does not** define all possible instances of these procedure inputs and outputs. This standard defines only those instances that may apply to any SCSI device or to processor type SCSI devices. Instances of the procedure inputs and outputs that apply to specific SCSI device models are defined in the applicable SCSI command standards (see 3.1.11).

This standard references values returned via the Status output parameter. Examples of such status values are CHECK CONDITION and COMMAND TERMINATED. Status values are **not** defined by this standard. SAM defines all Status values.

The entity that makes the procedure call from an initiator is an application client, as defined in SAM. The procedure call's representation arrives at the target in the form of a device service request. The entity that performs the work of the procedure call in a target is a device server, which is an object within a logical unit and is defined in SAM.

4.2 The Command Descriptor Block (CDB)

A command is communicated by sending a command descriptor block to the device server. For several commands, the command descriptor block is accompanied by a list of parameters in the Data-Out Buffer. See the specific commands for detailed information.

The command descriptor block shall have an operation code as its first byte and a control byte as its last byte. The general structure of the operation code and control byte are defined in SAM.

For all commands, if there is an invalid parameter in the command descriptor block, then the device server shall terminate the command without altering the medium.

Table 1 shows the typical format of a 6-byte CDB. Table 2 shows the typical format of a 10-byte CDB. Table 3 shows the typical format of a 12-byte CDB. Table 4 shows the typical format of a 16-byte CDB.

The following field descriptions apply to tables 1, 2, 3, and 4. Operation code is the code value identifying the operation being requested by the CDB. SAM defines the general structure of the operation code value. This standard specifies the operation code values used by the commands defined herein. In the typical usage, the Logical block address field contains a logical block address (see SBC). In the typical usage, the Transfer length field specifies the number of bytes, logical blocks,

or other command-specific units to be transferred. In the typical usage, the Parameter list length field specifies the number bytes of command parameter data to be sent from the application client to the device server. In the typical usage, the Allocation length field specifies the number of bytes set aside by the application client to receive command parameter data from the device server. The contents of the control field are defined in SAM.

Only the operation code and control fields have consistently defined meanings across all commands. The field uses shown in tables 1, 2, 3, and 4 are used consistently by most commands. However, the actual usage of any field (except operation code and control) is described in the clause defining that command.

Table 1 − Typical CDB for 6-byte commands

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

Table 2 − Typical CDB for 10-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code							
1	Reserved							
2	(MSB)							
3	Logical block address (if required)							
4								
5								(LSB)
6	Reserved							
7	(MSB)			Transfer length (if required) Parameter list length (if required)				
8				Allocation length (if required)				(LSB)
9	Control							

Table 3 – Typical CDB for 12-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code							
1	Reserved							
2	(MSB)							
3	Logical block address (if required)							
4								
5								(LSB)
6	(MSB)							
7	Transfer length (if required) Parameter list length (if required) Allocation length (if required)							
8								
9								(LSB)
10	Reserved							
11	Control							

Table 4 – Typical CDB for 16-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code							
1	Reserved							
2	(MSB)							
3	Logical block address (if required)							
4								
5								(LSB)
6	(MSB)							
7	Additional CDB data (if required)							
8								
9								(LSB)
10	(MSB)							
11	Transfer length (if required) Parameter list length (if required) Allocation length (if required)							
12								
13								(LSB)
14	Reserved							
15	Control							

4.2.1 Logical block address

The logical block address on logical units or within a partition on device volumes shall begin with block zero and be contiguous up to the last logical block on that logical unit or within that partition.

A six-byte command descriptor block contains a 21-bit logical block address. The ten-byte, the twelve-byte and the sixteen-byte command descriptor blocks contain 32-bit logical block addresses. Logical block addresses in additional parameter data have their length specified for each occurrence. See the specific command descriptions.

4.2.2 Transfer length

The transfer length field specifies the amount of data to be transferred, usually the number of blocks. For several commands the transfer length indicates the requested number of bytes to be sent as defined in the command description. For these commands the transfer length field may be identified by a different name. See the following descriptions and the individual command descriptions for further information.

Commands that use one byte for the transfer length allow up to 256 blocks of data to be transferred by one command. A transfer length value of 1 to 255 indicates the number of blocks that shall be transferred. A value of zero indicates 256 blocks.

In commands that use multiple bytes for the transfer length, a transfer length of zero indicates that no data transfer shall take place. A value of one or greater indicates the number of blocks that shall be transferred.

Refer to the specific command description for further information.

4.2.3 Parameter list length

The parameter list length is used to specify the number of bytes sent from the data-out buffer. This field is typically used in command descriptor blocks for parameters that are sent to a device server (e.g., mode parameters, diagnostic parameters, log parameters, etc.). A parameter length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

4.2.4 Allocation length

The allocation length field specifies the maximum number of bytes that an application client has allocated for returned data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The device server shall terminate transfers to the data-in buffer when allocation length bytes have been transferred or when all available data have been transferred, whichever is less. The allocation length is used to limit the maximum amount of data (e.g., sense data, mode data, log data, diagnostic data, etc.) returned to an application client.

5 Model common to all device types

This model describes some of the general characteristics expected of most SCSI devices. It is not intended to alter any requirements defined elsewhere in SCSI-3. Devices conforming to this standard also shall conform to the SCSI-3 Architecture Model (SAM).

5.1 Commands implemented by all SCSI device servers

This standard defines four commands that all SCSI-3 device servers shall implement - INQUIRY, REQUEST SENSE, SEND DIAGNOSTIC, and TEST UNIT READY. These commands are used to configure the system, to test devices, and to return important information concerning errors and exception conditions.

5.1.1 Using the INQUIRY command

The INQUIRY command may be used by an application client to determine the configuration of the logical unit. Device servers respond with information that includes their type and standard version and may include the vendor’s identification, model number and other useful information. It is recommended that device servers be capable of returning this information (or whatever part of it that is available) upon completing power-on initialization. A device server may take longer to get certain portions of this information, especially if it retrieves the information from the medium.

5.1.2 Using the REQUEST SENSE command

Whenever a command completes with a CHECK CONDITION or COMMAND TERMINATED status and Autosense Data is not provided, the application client that received the error status should issue a REQUEST SENSE command to receive the sense data describing the what cause of the condition. If the application client issues a command other than REQUEST SENSE, the sense data is lost.

5.1.3 Using the SEND DIAGNOSTIC command

The SEND DIAGNOSTIC command provides a means to request that an SCSI device perform a self test. While the test is device specific, the means of requesting the test is standardized. The response is simply a GOOD status if the test is successful or a CHECK CONDITION status if the test fails.

The SEND DIAGNOSTIC command also provides other optional features when used in conjunction with the RECEIVE DIAGNOSTIC RESULTS command.

5.1.4 Using the TEST UNIT READY command

The TEST UNIT READY command allows an application client to poll a logical unit until it is ready without the need to allocate space for returned data. It is especially useful to check the cartridge status of logical units with removable media. Device servers should respond promptly to indicate the current status of the device, delays to achieve good status are not advisable.

5.2 Parameter rounding

Certain parameters sent to a device server with various commands contain a range of values. Device servers may choose to implement only selected values from this range. When the device server receives a value that it does not support, it either rejects the command (CHECK CONDITION status with ILLEGAL REQUEST sense key) or it rounds the value received to a supported value. The device server shall reject unsupported values unless rounding is permitted in the description of the parameter.

When parameter rounding is implemented, a device server that receives a parameter value that is not an exact supported value shall adjust the value to one that it supports and shall return CHECK CONDITION status with a sense key of RECOVERED ERROR. The additional sense code shall be set to ROUNDED PARAMETER. The application client should issue an appropriate command to learn what value the device server has selected.

NOTE 1 Generally, the device server should adjust maximum-value fields down to the next lower supported value than the one specified by the application client. Minimum-value fields should be rounded up to the next higher supported value than the one specified by the application client. In some cases, the type of rounding (up or down) is explicitly specified in the description of the parameter.

5.3 Reservations

Commands that establish reservations may be used to restrict the execution of commands to a logical unit or a portion of the logical unit. Using the reservation commands, application clients may procure assistance from the device server to share and protect data or resources. If the application clients do not cooperate in the execution of a reservation protocol, data may be unexpectedly modified and deadlock conditions may occur.

This clause provides a general overview of reservations. The general description of reservations involves two groups of considerations; a) the type of reservation established, and b) the method used to establish, rescind, and manage the reservation. There are limits on the combinations of reservation types available under some reservation management methods. See the reservations management commands descriptions for details.

The types of reservations that can be established are:

a) **logical unit reservations** - a logical unit reservation restricts access to the entire logical unit;
b) **extent reservations** - an extent reservation restricts access to a specified extent within a logical unit; and
c) **element reservations** - an element reservation restricts access to a specified element within a medium changer.

The types of reservations can be further qualified by restrictions on types of access (e.g., read, write, control, etc.). However, access type restrictions are handled as an aspect of reservation management, not as an aspect of the type of reservation being established. In addition, some methods of reservations management permit establishing reservations on behalf of another device in the same SCSI domain (third-party reservations).

The methods of managing reservations are identified by the commands associated with the methods. The methods of managing reservations are:

a) Reserve/Release - associated with the RESERVE(6), RELEASE(6), RESERVE(10), and RELEASE(10) commands (see 7.22, 7.18, 7.21, and 7.17); and
b) Persistent Reservations - associated with the PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands (see 7.13 and 7.12).

The reservation restrictions placed on commands that explicitly read or write the medium result from the type of reservation combined with any access qualifications. The details of the reading and writing restrictions are described in this standard in the clauses that define the commands associated with each management method. For the Reserve/Release management method, see 7.22. For the Persistent Reservations management method, see 7.13.

For commands that do not explicitly read or write the medium, the applicable reservation restrictions depend solely on the type of reservations that are established at the time the command reaches the device server. Access qualifications (if any) and the reservations management method used to establish the reservations have no interaction with the restrictions placed on commands that do not explicitly read or write the medium. However, the particular reservation restrictions imposed are highly dependent on the relationship between the command and the type of reservations established when that command arrives at the device server. Therefore, the reservation restrictions for commands that do not explicitly access the medium are defined in the device model clause or in the clause defining that specific command.

The commands that manage reservations may be thought of as a special group within the group of commands that do not explicitly read or write the medium. The clauses defining the reservations management commands contain definitions of the interactions between them, which forms the overall reservations management paradigm. (See 7.12, 7.13, 7.18, 7.17, 7.22, and 7.21.)

Because a device server is unable to differentiate among the reservations made by different application clients running on an initiator, all application clients on the initiator have the same access restrictions. When multiple application clients are accessing a single device server from one initiator, the application clients should coordinate reservations and persistent reservations.

5.3.1 Reservation conflicts

A reservation conflict occurs when a device server receives a command that is prohibited from execution by an established reservation. The device server shall test for reservation conflicts at the time when a task enters the enabled task state. If a reservation conflict precludes any part of the command, none of the command shall be performed. When a reservation conflict is detected, the device server shall terminate that command with a RESERVATION CONFLICT status.

For each command, this standard or a related command standard (see 3.1.11) defines the conditions that result in RESERVATION CONFLICT. The conditions are identified as part of the device model or command definition.

5.3.2 The Reserve/Release management method

The Reserve/Release management method commands, RESERVE(6), RESERVE(10), RELEASE(6), and RELEASE(10) are used among multiple initiators that do not require operations to be protected across initiator failures (and subsequent hard resets). The Reserve/Release reservations management method also allows an application client to provide restricted device access to one additional initiator (a third-party initiator), usually a temporary initiator performing a service for the application client sending the reservation command.

Reservations managed using the Reserve/Release method do not persist across some recovery actions (e.g., hard resets), so most systems require significant reinitialization after a failure that results in a hard reset. Reserve/Release managed reservations are retained by the device server until released or until reset by mechanisms specified in this standard.

The RESERVE(6) and RESERVE(10) commands allow superseding reservations.

5.3.3 The Persistent Reservations management method

The Persistent Reservations management method is used among multiple initiators that require operations to be protected across initiator failures, which usually involve hard resets. Persistent reservations persist across recovery actions, to provide initiators with more detailed control over reservations recovery. Persistent reservations for failing initiators may be preempted by another initiator as part of the recovery process. Persistent reservations are retained by the device server until released, preempted, or until cleared by mechanisms specified in this standard. Persistent reservations are optionally retained when power to the target is lost.

5.4 Multiple port and multiple initiator behavior

SAM specifies the behavior of logical units being accessed by more than one initiator. Additional ports provide alternate service delivery paths through which the device server may be reached and may also provide connectivity for additional initiators. An alternate path may be used to improve the availability of devices in the presence of certain types of failures and to improve the performance of devices whose other paths may be busy.

If a target has more than one service delivery port, the arbitration and connection management among the ports is defined by the implementation. Above the interconnect implementation, two contention resolution options exist:

a) If one port on a target is being used by an initiator, accesses attempted through another port may receive a status of BUSY; or
b) If the target has sufficient internal resources, commands may be accepted through other ports while one port is being used.

The device server shall indicate the presence of multiple ports by setting the MultiP bit to 1 in its standard INQUIRY data.

Once a device server grants a reservation, all initiators (regardless of port) except the initiator to which the reservation was granted shall be treated as different initiators. Only the following operations allow an initiator to interact with the tasks of another initiator, regardless of the service delivery port:

a) the PERSISTENT RESERVE OUT with Preempt service action removes persistent reservations for another initiator (see 7.13.1.5);
b) the PERSISTENT RESERVE OUT with Preempt and Clear service action removes persistent reservations and all tasks for another initiator (see 7.13.1.6);
c) the PERSISTENT RESERVE OUT with Clear service action removes persistent reservations and reservation keys for all initiators (see 7.13.1.4);
d) the TARGET RESET task management function removes reservations established by the Reserve/Release method and removes all tasks for all logical units in the target and for all initiators (see SAM). Persistent reservations remain unmodified;
e) the LOGICAL UNIT RESET task management function removes reservations established by the Reserve/Release method and removes all tasks for all initiators for the addressed logical unit and any logical units depending from it in a hierarchical addressing structure (see SAM). Persistent reservations remain unmodified; and
f) the CLEAR TASK SET task management function removes all tasks for the selected logical unit for all initiators. Most other machine states remain unmodified, including MODE SELECT parameters, reservations, and ACA (see SAM).

5.5 Removable medium devices with an attached medium changer

When a logical unit is served by a medium changer, control over one medium transport element may be effected using medium changer commands sent to the device server within the logical unit. The level of control is not as complete as would be available if a fully functional medium-changer device server were implemented (see SMC). However, the amount of control is sufficient for paired device and medium changer configurations.

The device server shall indicate its ability to support medium changer commands by setting the MChngr bit to one in its standard INQUIRY data (see 7.5.1). An MChngr bit of one shall indicate that the MOVE MEDIUM ATTACHED and READ ELEMENT STATUS ATTACHED commands are supported by the device server. Definitions of the MOVE MEDIUM ATTACHED and READ ELEMENT STATUS ATTACHED commands may be found in SMC.

Only one medium transport element shall be permitted, element 0. Only one data transfer element shall be permitted. Media exchanges shall not be supported by attached medium changers. The RESERVE ELEMENT and RELEASE ELEMENT commands shall not be supported by attached medium changers.

6 Model for processor devices

An SCSI processor device is a primary computing device with the characteristics of a device server, typically a personal computer, minicomputer, mainframe computer, or auxiliary computing device or server. Such a primary computing device is often called a host. The processor device receives or provides packets of data as requested by an application client.

In the SCSI processor device, the device server accepts and provides data packets transferred according to commands from the application client. An application client and the processor device server are assumed to have a common set of rules by which information to be is exchanged between them, how the information is interpreted by the processor device server, and when it is allowable to exchange the information. These rules are not specified by this standard.

The application client requests that the processor device server accept a packet of data by transmitting a SEND command. The application client requests that the processor device server return a packet of data by transmitting a RECEIVE command. A COPY command may also be transmitted to the processor device server to request that it serve as a copy manager. The data flow may be between the processor device and another SCSI device or may be between two SCSI devices under control of the processor device acting as a third-party copy manager.

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

a) Terminate the command with CHECK CONDITION status and the sense key NOT READY with the appropriate additional sense code for the condition. This is the appropriate response to a TEST UNIT READY command;
b) Delay data transmission until the necessary resource or data packet becomes available;
c) Terminate the command with BUSY status; or
d) Treat the logical unit as an incorrect logical unit (see SAM).

A single target may have more than one logical unit. Logical units may serve as additional paths to a single resource, and/or each logical unit may serve as a path to different resources within the device. A single logical unit may also serve as a path to multiple resources if the processor device server may interpret information within the data packet and route the packet to the appropriate resource.

If the processor device server determines that an error or unusual condition has occurred while performing an operation specified by the contents of a data packet, the information describing the condition is returned as a part of a data packet. If the processor device server determines that an error or unusual condition has occurred while executing the SCSI command

from the application client, the command is terminated with a CHECK CONDITION and the failures are identified through the sense data.

The SCSI processor device is distinguished from an SCSI communications device by the fact that the primary destination of the data packets is within the target device. An SCSI communications device, passes the data on to an ultimate destination outside the target through a network. Many types of devices may function as processor devices if no other suitable SCSI device type exists and if the packet exchange protocol dictated by the processor device model meets their functional requirements.

Processor device types shall not implement extent or element reservations.

7 Commands for all device types

The operation codes for commands that apply to all device types are listed in table 5.

Table 5 — Commands for all device types

Command name	Operation code	Type	Clause
CHANGE DEFINITION	40h	O	7.1
COMPARE	39h	O	7.2
COPY	18h	O	7.3
COPY AND VERIFY	3Ah	O	7.4
INQUIRY	12h	M	7.5
LOG SELECT	4Ch	O	7.6
LOG SENSE	4Dh	O	7.7
MODE SELECT(6)	15h	Z	7.8
MODE SELECT(10)	55h	Z	7.9
MODE SENSE(6)	1Ah	Z	7.10
MODE SENSE(10)	5Ah	Z	7.11
MOVE MEDIUM ATTACHED **[1]**	A7h	Z	SMC
PERSISTENT RESERVE IN	5Eh	Z	7.12
PERSISTENT RESERVE OUT	5Fh	Z	7.13
PREVENT ALLOW MEDIUM REMOVAL	1Eh	O	7.14
READ BUFFER	3Ch	O	7.15
READ ELEMENT STATUS ATTACHED **[1]**	B4h	Z	SMC
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	7.16
RELEASE(6)	17h	Z	7.17
RELEASE(10)	57h	Z	7.18
REPORT LUNS	A0h	O	7.19
REQUEST SENSE	03h	M	7.20
RESERVE(6)	16h	Z	7.21
RESERVE(10)	56h	Z	7.22
SEND DIAGNOSTIC	1Dh	M	7.23
TEST UNIT READY	00h	M	7.24
WRITE BUFFER	3Bh	O	7.25

Key: M = Command implementation is mandatory.
O = Command implementation is optional.
Z = Command implementation is device type specific.

Notes:
[1] The MOVE MEDIUM ATTACHED and READ ELEMENT STATUS ATTACHED operation codes shown here should be used by devices with attached medium changers.

7.1 CHANGE DEFINITION command

The CHANGE DEFINITION command (see table 6) is used to modify the operating definition of the device server(s) with respect to commands from the sending initiator or with respect to commands from all initiators.

Table 6 — CHANGE DEFINITION command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (40h)							
1	Reserved							
2	Reserved							Save
3	Reserved	Definition parameter						
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							
8	Parameter data length							
9	Control							

If reservations are active, they shall affect the execution of the CHANGE DEFINITION command as follows. If the SCSI device does not allow different operating definitions for each initiator, a reservation conflict shall occur when a CHANGE DEFINITION command is received from an initiator other than the one holding a logical unit reservation. If any initiator has an extent or element reservation on an SCSI device, no other initiator may affect the operating definition of the initiator holding the reservation by use of the CHANGE DEFINITION command.

A save control bit (Save) of zero indicates that the device server shall not save the operating definition. A Save bit of one indicates that the device server shall save the operating definition in non-volatile memory.

The definition parameter field is defined in table 7.

Table 7 — Definition parameter field

Value	Meaning of definition parameter
00h	Use current operating definition
03h	SCSI-2 operating definition
04h	SCSI-3 operating definition
01 - 02h	Reserved for historical uses
05 - 3Eh	Reserved
3Fh	Manufacturer default definition
40 - 7Fh	Vendor-specific

The current operating definition parameter values establish operating definitions compatible with the applicable SCSI standard. Definitions supported by an SCSI device are returned in the implemented operating definition page (see 8.4.4).

The parameter data length field specifies the length in bytes of the parameter data that shall be transferred from the application client to the device server. A parameter data length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. Parameter data length values greater than zero indicate the number of bytes of parameter data that shall be transferred.

The parameter data is vendor-specific.

NOTE 2 The parameter data may be used to specify a password to validate an operating definition change.

The CHANGE DEFINITION command causes one of the operating definition modifications listed below:

a) Change the operating definition of a logical unit relative to the initiator that issued the command: In this case, the target is capable of maintaining a separate operating definition for each logical unit relative to each initiator in the system;
b) Change the operating definition of all logical units in the target relative to the initiator that issued the command: In this case, the target is capable of maintaining a unique operating definition, for each initiator in the system, that applies to all logical units in the target;
c) Change the operating definition of a logical unit relative to all initiators in the system: In this case, the target is capable of maintaining a separate operating definition for each logical unit relative to all initiators in the system; or
d) Change the operating definition of all logical units in the target relative to all initiators in the system: In this case, the target is capable of maintaining only one operating definition.

NOTES
3 This standard does not provide a direct means to determine which of the above four methods has been implemented. An indirect means of determining which method is implemented exists in that the device server is required to inform affected initiators of operating definition changes via the unit attention condition.
4 The modifications listed c) and d) above may result in incompatibilities if other initiators are using a different SCSI version.

The operating definition is modified after successful completion of the command. The application client should verify the new operating definition by issuing an INQUIRY command requesting the implemented operating definition page (see 8.4.1).

NOTE 5 The method of changing the operating definition is vendor-specific. Some implementations may require that the target's operating mode be reinitialized as if a power-up or hard reset had occurred. Other implementations may modify only those operating definitions that are affected by the CHANGE DEFINITION command.

If the CHANGE DEFINITION command is not executed successfully for any reason, the operating definition shall remain the same as it was before the CHANGE DEFINITION command was attempted. If it is impossible to return to the previous operating definition, a unit attention condition shall be generated.

NOTE 6 The present operating definition of the target may always be interrogated through the INQUIRY command. When an SCSI-3 target has its operating definition changed to an older SCSI operating definition, certain changes are needed to promote compatibility with preexisting older SCSI initiators.

After a power-on condition or a hard reset condition, the target shall set its initial operating definition of the device server(s) to the last saved value (if saving is implemented), or its default value (if saving is not implemented).

7.2 COMPARE command

The COMPARE command (see table 8) provides the means to compare data from one logical unit with another or the same logical unit in a manner similar to the COPY command.

Table 8 – COMPARE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (39h)							
1	Reserved							Pad
2	Reserved							
3	(MSB)							
4	Parameter list length							
5								(LSB)
6	Reserved							
7	Reserved							
8	Reserved							
9	Control							

If reservations are active, they shall affect the execution of the COMPARE command as follows. A reservation conflict shall occur when a COMPARE command is received from an initiator other than the one holding a logical unit reservation. The COMPARE command shall be evaluated for extent reservation conflicts as if the copy master were performing normal read operations even when an SCSI device is requested to compare with itself. For example, if a COMPARE is issued to logical unit 0 that requests the SCSI device to compare between data from logical unit 0 to data from logical unit 1, access to logical unit 1 also shall be evaluated for a reservation conflict. COMPARE commands shall be terminated with CHECK CONDITION status and the sense key shall be set to DATA PROTECT if any part of the compare operation is prohibited by an extent reservation.

This command functions in the same manner as the COPY command, except that the data from the source is compared on a byte-by-byte basis with the data from the destination. All fields in the COMPARE command CDB have the same meaning as the equivalent fields in the COPY command CDB. The parameter list transferred to the target is the same as for the COPY command. This parameter list contains the information to identify the logical units involved in the comparison and the length of the comparison. See 7.3 for information about the COPY command.

If the comparison is unsuccessful, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to MISCOMPARE. The remaining fields in the sense data shall be set as documented in the COPY command.

7.3 COPY command

The COPY command (see table 9) provides a means to copy data from one logical unit to another or the same logical unit. The device server that receives and performs the COPY command is called the copy manager. The copy manager is responsible for copying data from a logical unit (source device) to a logical unit (destination device). These logical units may reside on different SCSI devices or the same SCSI device (in fact all three may be the same logical unit). Device servers that implement this command are not required to support copies to or from another SCSI device, and are not required to support third party copies (i.e., both the source and the destination logical units reside on other SCSI devices).

Table 9 — COPY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (18h)							
1	Reserved							Pad
2	(MSB)							
3	Parameter list length							
4								(LSB)
5	Control							

If reservations are active, they shall affect the execution of the COPY command as follows. A reservation conflict shall occur when a COPY command is received from an initiator other than the one holding a logical unit reservation. The COPY command shall be evaluated for extent reservation conflicts as if the copy master were performing normal write and read operations even when an SCSI device is requested to copy to or from itself. For example, if a COPY is issued to logical unit 0 that requests the SCSI device to copy data from logical unit 0 to logical unit 1, access to logical unit 1 also shall be evaluated for a reservation conflict. 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 extent reservation.

The Pad bit is used in conjunction with the Cat bit (see 7.3.7) in the segment descriptors to define what action should be taken when a segment of the copy does not fit exactly into an integer number of destination blocks.

The parameter list length field specifies the length in bytes of the parameters that shall be contained in the Data-Out Buffer. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The COPY parameter list (see table 10) begins with a four-byte header that contains the COPY function code and priority. Following the header is one or more segment descriptors.

Table 10 – COPY parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	COPY function code					Priority		
1	Vendor-specific							
2	Reserved							
3	Reserved							
	Segment descriptor(s)							
0	Segment descriptor 0							
n	(See specific table for length.)							
	.							
0	Segment descriptor x							
n	(See specific table for length.)							

The COPY function code field defines a specific format for the segment descriptors. A copy may be divided into multiple segments. Each segment shall be described by a segment descriptor. All segment descriptors shall have the format specified by the COPY function code. Table 11 defines the COPY function codes, identifies the table showing the required segment descriptor format for each COPY function code, and provides other information about each COPY function code. A device server need not support all function codes for its device type.

Table 11 – COPY function codes

Peripheral device type (Note 6)		**COPY function code**	**Segment descriptor table**	**Comments**
Source	**Destination**			
Block devices (Device types 0,4,5,7)	Stream devices (Device types 1,2,3,9)	0Ah	12	
Stream devices (Device types 1,3,9)	Block devices (Device types 0,4,5,7)	0Bh	12	(Note 5)
Block devices (Device types 0,4,5,7)	Block devices (Device types 0,4,5,7)	0Ch	13	(Note 5)
Stream devices (Device types 1,3,9)	Stream devices (Device types 1,2,3,9)	0Dh	14	
Sequential-access (Device type 1)	Sequential-access (Device type 1)	0Eh	15	Image copy

NOTES
1 COPY function code 0Fh is reserved for future standardization.
2 COPY function codes 00h - 04h are defined in the SCSI-2 Standard.
3 COPY function codes 05h - 09h are reserved.
4 COPY function codes 10h - 1Fh are vendor-specific.
5 When using the COMPARE command the destination block device may be a CD-ROM device or an optical-memory device that uses read-only media.
6 See 7.5.1 for peripheral device type definitions.

The priority field of the COPY parameter list establishes the relative priority of this COPY command to other commands being executed by the same device server. All commands that do not have a COPY parameter list (see table 10) are assumed to have a priority of 1. Priority 0 is the highest priority, with increasing priority values indicating lower priorities.

The segment descriptor formats are determined by the COPY function code. The segment descriptor format used for block devices (i.e., write-once, CD-ROM, optical-memory, and direct-access devices) shall be the same. The segment descriptor format used for stream devices (i.e., printer, processor, communications, and sequential-access devices) shall be the same. Thus a copy operation from a write-once device to a printer device uses the same segment descriptor format as a copy operation from a direct-access device to a sequential-access device (see table 11). The segment descriptor formats are described in 7.3.3 through 7.3.6. A maximum of 256 segment descriptors are permitted. The segment descriptors are identified by ascending numbers beginning with zero.

7.3.1 Errors detected by the copy manager

Two classes of exception conditions may occur during execution of a COPY command. The first class consists of those exception conditions detected by the copy manager. These conditions include parity errors while transferring the COPY command and status byte, invalid parameters in the COPY command, invalid segment descriptors, and inability of the copy manager to continue operating. In the event of such an exception condition, the copy manager shall:

a) terminate the COPY command with CHECK CONDITION status;
b) set the valid bit in the sense data to one. The segment number shall contain the number of the segment descriptor being processed at the time the exception condition is detected. The sense key shall contain the sense key code describing the exception condition (i.e., not COPY ABORTED). The information field shall contain the difference between the number of blocks field in the segment descriptor being processed at the time of the failure and the number of blocks successfully copied. This number is the residue of unprocessed blocks remaining for the segment descriptor.

7.3.2 Errors detected by a target servicing a copy manager

The second class of errors consists of exception conditions detected by the SCSI device transferring data at the request of a copy manager. The copy manager detects exception conditions by receiving CHECK CONDITION status from one of the SCSI devices it is managing. It then shall recover the sense data associated with the exception condition. After recovering the sense data, the copy manager shall clear the ACA associated with the CHECK CONDITION status.

The copy manager may also be the source or destination SCSI device (or both). It shall distinguish between a failure of the management of the COPY and a failure of the data transfer being requested. It shall then create the appropriate sense data and manage the ACA condition without intervention from the original application client.

After recovering the sense data and clearing the ACA condition associated with the detected error, the copy manager shall:

a) terminate the COPY command with CHECK CONDITION status;
b) the valid bit in the sense data shall be set to one. The segment number shall contain the number of the segment descriptor being processed at the time the exception condition is detected. The sense key shall be set to COPY ABORTED. The information field shall contain the difference between the number of blocks field in the segment descriptor being processed at the time of the failure and the number of blocks successfully copied. This number is the residue of unprocessed blocks remaining for the segment descriptor.

The first byte of the command-specific information field in the sense data shall specify the starting byte number, relative to the first byte of sense data, of an area that contains (unchanged) the source logical unit's status byte and sense data. A zero value indicates that no status byte or sense data is being returned for the source logical unit.

The second byte of the command-specific information field in the sense data shall specify the starting byte number, relative to the first byte of sense data, of an area that contains (unchanged) the destination logical unit's status byte and sense data. A zero value indicates that no status byte or sense data is being returned for the destination logical unit.

7.3.3 COPY function codes 0Ah and 0Bh

The format for the segment descriptors for COPY transfers between block and stream devices is specified in table 12. This format is required for COPY function codes 0Ah and 0Bh. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 12 – Segment descriptor for COPY function codes 0Ah and 0Bh

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved				Cat	Reserved		
1	Reserved							
2	(MSB)							
9	Source target identifier							(LSB)
10	(MSB)							
17	Source Logical Unit Number							(LSB)
18	(MSB)							
25	Destination target identifier							(LSB)
26	(MSB)							
33	Destination Logical Unit Number							(LSB)
34	(MSB)							
35	Stream device block length							(LSB)
36	(MSB)							
39	Block device number of blocks							(LSB)
40	(MSB)							
43	Block device logical block address							(LSB)

The Source target identifier and Source Logical Unit Number fields specify the SCSI target identifier and logical unit to copy the data from for this segment of the COPY command. The Destination target identifier and Destination Logical Unit Number fields specify the SCSI target identifier and logical unit to copy the data to for this segment of the COPY command. Definitions of the format of target identifiers and Logical Unit Numbers are protocol-specific. Device servers are not required to support third-party COPY in which the copy manager is not the source or destination device. Some device servers only support COPY within the SCSI device and not to other SCSI devices. If an unsupported COPY operation is requested, the 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 PARAMETER LIST (see 7.3.1).

The Cat bit is used in conjunction with the Pad bit (see 7.3.7) in the segment descriptors to define what action should be taken when a segment of the copy does not fit exactly into an integer number of destination blocks.

The stream device block-length field specifies the block length to be used on the stream device logical unit during this segment of the COPY command. If the copy manager detects that this block length is not supported, the 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 PARAMETER LIST. If the block length is found to be invalid while executing a read or write operation to the stream device, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to COPY ABORTED (see 7.3.2).

The block device number of blocks field specifies the number of blocks in the current segment to be copied. A value of zero indicates that no blocks shall be transferred in this segment.

The block device logical block address field specifies the starting logical block address on the logical unit for this segment.

7.3.4 COPY function code 0Ch

The format for the segment descriptors for COPY transfers among block devices is specified in table 13. This format is required for COPY function code 0Ch. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 13 – Segment descriptor for COPY function code 0Ch

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved			DC	Cat	Reserved		
1	Reserved							
2	(MSB)			Source target identifier				
9								(LSB)
10	(MSB)			Source Logical Unit Number				
17								(LSB)
18	(MSB)			Destination target identifier				
25								(LSB)
26	(MSB)			Destination Logical Unit Number				
33								(LSB)
34	Reserved							
35	Reserved							
36	(MSB)			Number of blocks				
39								(LSB)
40	(MSB)			Source logical block address				
43								(LSB)
44	(MSB)			Destination logical block address				
47								(LSB)

See 7.3.3 for definitions of the Source target identifier, the Source Logical Unit Number, the Destination target identifier, the Destination Logical Unit Number, and Cat fields.

A destination count (DC) bit of zero indicates that the number of blocks field refers to the source logical unit. A DC bit of one indicates that the number of blocks field refers to the destination logical unit.

The Number of blocks field specifies the number of blocks to be transferred to or from (depending on the DC bit) the block device during this segment. A value of zero indicates that no blocks shall be transferred.

The Source logical block address field specifies the starting logical block address on the source block device.

The Destination logical block address field specifies the starting logical block address on the destination block device.

7.3.5 COPY function code 0Dh

The format for the segment descriptors for COPY transfers among stream devices is specified by table 14. This format is required for COPY function code 0Dh. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 14 − Segment descriptor for COPY function code 0Dh

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved			DC	Cat	Reserved		
1	Reserved							
2	(MSB)			Source target identifier				
9								(LSB)
10	(MSB)			Source Logical Unit Number				
17								(LSB)
18	(MSB)			Destination target identifier				
25								(LSB)
26	(MSB)			Destination Logical Unit Number				
33								(LSB)
34	Reserved							
35	Reserved							
36	(MSB)			Source block length				
37								(LSB)
38	(MSB)			Destination block length				
39								(LSB)
40	(MSB)			Number of blocks				
43								(LSB)

See 7.3.3 for definitions of the Source target identifier, the Source Logical Unit Number, the Destination target identifier, the Destination Logical Unit Number, and Cat fields.

A destination count (DC) bit of zero indicates that the number of blocks field refers to the source logical unit. A DC bit of one indicates that the number of blocks field refers to the destination logical unit.

The Source block length field specifies the block-length of the source device for this segment of the COPY. A zero in this field indicates variable block-length. For non-zero values, this field shall match the logical unit's actual block-length.

If block-length mismatches are detected prior to the beginning of the read operation by the copy manager, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST (see 7.3.1).

If the mismatches are detected during the read operation by the copy manager, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to COPY ABORTED (see 7.3.2). The additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

The Destination block-length field specifies the block length to be used on the destination logical unit during the COPY. Destination block length mismatches are handled in an analogous manner as source block length mismatches.

The Number of blocks field specifies the number of blocks to be transferred to or from (depending on the DC bit) the device during this segment. A value of zero indicates that no blocks shall be transferred.

7.3.6 COPY function code 0Eh

The format for the segment descriptors for image COPY transfers between sequential-access devices is specified in table 15. This format is required for COPY function code 0Eh. The segment descriptor may be repeated up to 256 times within the parameter list length specified in the command descriptor block.

Table 15 − Segment descriptor for COPY function code 0Eh

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							
2	(MSB)			Source target identifier				
9								(LSB)
10	(MSB)			Source Logical Unit Number				
17								(LSB)
18	(MSB)			Destination target identifier				
25								(LSB)
26	(MSB)			Destination Logical Unit Number				
33								(LSB)
34	Count							
35	Reserved							
39								
40	Vendor-specific							
43								

See 7.3.3 for definitions of the Source target identifier, the Source Logical Unit Number, the Destination target identifier and the Destination Logical Unit Number.

The image COPY function copies an exact image of the source device medium to the destination device medium, beginning at their current positions. The copy function terminates when the source device:

a) encounters an end-of-partition as defined by the source device;
b) encounters an end-of-data as defined by the source device (i.e., BLANK CHECK sense key);
c) has copied the number of consecutive filemarks specified in the count field from the source device to the destination device; or
d) has copied the number of consecutive setmarks specified in the count field from the source device to the destination device, if the RSmk bit in the device configuration page (see SSC) is one.

A count field of zero indicates that the COPY command shall not terminate due to any number of consecutive filemarks or setmarks. Other error or exception conditions (e.g., early-warning end-of-partition on the destination device) may cause the COPY command to terminate prior to completion. In such cases, it is not possible to calculate a residue, so the information field in the sense data shall be set to zero.

7.3.7 Copies with unequal block lengths

When copying data between two devices with unequal block lengths, it is possible for the last source block to not completely fill the last destination block for one or more segments in the COPY command. Two optional bits are defined to assist in controlling the copy manager's actions in this circumstance. The Pad bit (in the command descriptor block) and the Cat bit (in each applicable segment descriptor) are defined in table 16.

Table 16 – Pad and Cat bit definition

Pad	Cat	COPY manager's action
0	0	On inexact segments, it is device specific whether the copy manager rejects the COPY command with CHECK CONDITION status and ILLEGAL REQUEST sense key, the copy manager writes or accepts short blocks (variable-block mode on sequential-access devices), or the copy manager adds pad characters (00h) to the destination block or strips pad characters from the source block.
1	0	On inexact segments, the copy manager shall add pad characters (00h) to the destination block to completely fill the block, or it shall strip pad characters from the source block, always stopping at the end of a complete block.
X	1	The copy manager shall always write or read complete blocks. On inexact segments, the remainder of the block contains data from the next segment. This code is invalid in the last segment of the COPY command.

NOTE 7 Use of pad characters is intended to assist in managing COPY commands between devices of different block lengths where partial-block residues may occur. The application client that issued the COPY command is responsible for management of these pad areas (i.e., remembering where they are). One possible method is to write the COPY parameter list information to the destination medium prior to issuing the COPY command for backup and to read this information prior to issuing the COPY command for restore.

7.4 COPY AND VERIFY command

The COPY AND VERIFY command (see table 17) performs the same function as the COPY command, except that a verification of the data written to the destination logical unit is performed after the data is written. The parameter list transferred to the device server is the same as for the COPY command. This parameter list contains the information to identify the logical units involved in the copy and the length of the copy. See 7.3 for information about the COPY command.

Table 17 – COPY AND VERIFY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (3Ah)							
1	Reserved						BytChk	Pad
2	Reserved							
3	(MSB)							
4	Parameter list length							
5								(LSB)
6	Reserved							
7	Reserved							
8	Reserved							
9	Control							

If reservations are active, they shall affect the execution of the COPY AND VERIFY command as follows. A reservation conflict shall occur when a COPY AND VERIFY command is received from an initiator other than the one holding a logical unit reservation. The COPY AND VERIFY command shall be evaluated for extent reservation conflicts as if the copy master were performing normal write and read operations even when an SCSI device is requested to copy to or from itself. For example, if a COPY is issued to logical unit 0 that requests the SCSI device to copy data from logical unit 0 to logical unit 1, access to logical unit 1 also shall be evaluated for a reservation conflict. COPY AND VERIFY 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 extent reservation.

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 comparison of data written on the destination medium and the data transferred from the source medium. If the comparison is unsuccessful for any reason, the copy manager shall return CHECK CONDITION status with the sense key set to MISCOMPARE. The remaining fields in the sense data shall be set as documented in the COPY command.

7.5 INQUIRY command

The INQUIRY command (see table 18) requests that information regarding parameters of the target and a component logical unit be sent to the application client. Options allow the application client to request additional information about the target and logical unit (see 7.5.3) or information about SCSI commands supported by the device server (see 7.5.4).

Table 18 – INQUIRY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (12h)							
1	Reserved						CmdDt	EVPD
2	Page or Operation code							
3	Reserved							
4	Allocation length							
5	Control							

The INQUIRY command shall not be affected by reservations or persistent reservations.

An enable vital product data (EVPD) bit of one specifies that the device server shall return the optional vital product data specified by the page code field. If the logical unit does not support vital product data and this bit is set to one, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

A command support data (CmdDt) bit of one specifies that the device server shall return the optional command support data specified by the operation code field. If the device server does not support returning command data and this bit is set to one, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB. Details of the command support data may be found in 7.5.4.

NOTE 8 An SCSI-3 application client may receive a CHECK CONDITION status response with the sense key set to ILLEGAL REQUEST upon sending an INQUIRY command with the CmdDt bit set to 1 to some SCSI-2 device servers, since this bit was reserved in SCSI-2.

If both the EVPD and CmdDt bits are zero, the device server shall return the standard INQUIRY data (see 7.5.1). If the page or operation code field is not zero when both EVPD and CmdDt are zero, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

If both the EVPD and CmdDt bits are one, the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

When the EVPD bit is one, the page or operation code field specifies which page of vital product data information the device server shall return (see 8.4).

When the CmdDt bit is one, the page or operation code field specifies the SCSI operation code for which device server shall return command support data (see 7.5.4).

The INQUIRY command shall return CHECK CONDITION status only when the device server cannot return the requested INQUIRY data.

If an INQUIRY command is received from an initiator with a pending unit attention condition (i.e., before the device server reports CHECK CONDITION status), the device server shall perform the INQUIRY command and shall not clear the unit attention condition (see SAM).

The INQUIRY data should be returned even though the device server is not ready for other commands. To minimize delays after a hard reset or power-up condition, the standard INQUIRY data should be available without incurring any media access delays. If the device server does store some of the INQUIRY data on the media, it may return zeros or ASCII spaces (20h) in those fields until the data is available from the media.

The INQUIRY data may change as the target executes its initialization sequence or in response to a CHANGE DEFINITION command. For example, the target may contain a minimum command set in its non-volatile memory and may load its final firmware from the media when it becomes ready. After the target has loaded the firmware, it may support more options and therefore return different supported options information in the INQUIRY data.

If the standard INQUIRY data changes for any reason, the device server shall generate a unit attention condition for all initiators (see SAM). The device server shall set the additional sense code to INQUIRY DATA HAS CHANGED.

NOTE 9 The INQUIRY command is typically used by an application client after a hard reset or power-up condition to determine the device types for system configuration.

7.5.1 Standard INQUIRY data

The standard INQUIRY data (see table 19) shall contain at least 36 bytes.

Table 19 — Standard INQUIRY data format

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	RMB	Reserved						
2	ISO/IEC version		ECMA version			ANSI version		
3	AERC	TrmTsk	NormACA	Reserved	Response data format			
4	Additional length (n-4)							
5	Reserved							
6	Reserved	EncServ	VS	MultiP	MChngr	ACKREQQ†	Addr32†	Addr16†
7	RelAdr	WBus32†	WBus16†	Sync†	Linked	TranDis†	CmdQue	VS
8 — 15	(MSB) Vendor identification (LSB)							
16 — 31	(MSB) Product identification (LSB)							
32 — 35	(MSB) Product revision level (LSB)							
36 — 55	Vendor-specific							
56 — 95	Reserved							
	Vendor-specific parameters							
96 — n	Vendor-specific							

Note: † The meanings of these bits are specific to SIP (see 7.5.2). For protocols other than SIP, these bits are reserved.

The peripheral qualifier and peripheral device-type fields identify the device currently connected to the logical unit. If the target is not capable of supporting a device on this logical unit, the device server shall set this field to 7Fh (peripheral qualifier set to 011b and peripheral device type set to 1Fh). The peripheral qualifier is defined in table 20 and the peripheral device type is defined in table 21.

Table 20 — Peripheral qualifier

Qualifier	Description
000b	The specified peripheral device type is currently connected to this logical unit. If the device server cannot determine whether or not a physical device is currently connected, it also shall use this peripheral qualifier when returning the INQUIRY data. This peripheral qualifier does not mean that the device is ready for access by the initiator.
001b	The device server is capable of supporting the specified peripheral device type on this logical unit. However, the physical device is not currently connected to this logical unit.
010b	Reserved
011b	The device server is not capable of supporting a physical device on this logical unit. For this peripheral qualifier the peripheral device type shall be set to 1Fh to provide compatibility with previous versions of SCSI. All other peripheral device type values are reserved for this peripheral qualifier.
1XXb	Vendor-specific

Table 21 — Peripheral device type

Code	Doc.	Description
00h	SBC	Direct-access device (e.g., magnetic disk)
01h	SSC	Sequential-access device (e.g., magnetic tape)
02h	SSC	Printer device
03h	SPC	Processor device
04h	SBC	Write-once device (e.g., some optical disks)
05h	MMC	CD-ROM device
06h	SGC	Scanner device
07h	SBC	Optical memory device (e.g., some optical disks)
08h	SMC	Medium changer device (e.g., jukeboxes)
09h	SSC	Communications device
0Ah - 0Bh		Defined by ASC IT8 (Graphic arts pre-press devices)
0Ch	SCC	Storage array controller device (e.g., RAID)
0Dh	SES	Enclosure services device
0Eh - 1Eh		Reserved
1Fh		Unknown or no device type

A removable medium (RMB) bit of zero indicates that the medium is not removable. A RMB bit of one indicates that the medium is removable.

The values in the ISO version and ECMA version fields are defined by the International Organization for Standardization and the European Computer Manufacturers Association, respectively.

The ANSI version field indicates the implemented version of this standard and is defined in table 22.

Table 22 – ANSI version

Code	Description
0h	The device does not claim conformance to any standard.
1h	Obsolete.
2h	The device complies to ANSI X3.131-1994 (SCSI-2).
3h	The device complies to this standard.
4h - 7h	Reserved

The asynchronous event reporting capability (AERC) bit indicates that the target supports the asynchronous event reporting capability as defined in SAM. The AERC bit is qualified by the peripheral device type field as follows:

a) Processor device-type definition: An AERC bit of one indicates that the processor device is capable of accepting asynchronous event reports. An AERC bit of zero indicates that the processor device does not support asynchronous event reports; or
b) All other device-types: This bit is reserved.

Details of the asynchronous event reporting support are protocol-specific.

A terminate task (TrmTsk) bit of one indicates that the device server supports the TERMINATE TASK task management function as defined in SAM. A value of zero indicates that the device server does not support the TERMINATE TASK task management function.

The Normal ACA Supported bit (NormACA) of one indicates that the device server supports setting the NACA bit to one in the Control Byte of the CDB (as defined in SAM). A NormACA bit of zero indicates that the device server does not support setting the NACA bit to one.

A response data format value of two indicates that the data shall be in the format specified in this standard. Response data format values less than two are obsolete. Response data format values greater than two are reserved.

The additional length field shall specify the length in bytes of the parameters. If the allocation length of the command descriptor block is too small to transfer all of the parameters, the additional length shall not be adjusted to reflect the truncation.

An Enclosure Services (EncServ) bit of one indicates that the device contains an embedded enclosure services component. See SES for details about enclosure services, including a device model for an embedded enclosure services device. An EncServ bit of zero indicates that the device does not contain an embedded enclosure services component.

A Multi Port (MultiP) bit of one shall indicate that this is a multi-port (2 or more ports) device and conforms to the SCSI-3 multi-port device requirements found in the applicable standards. A value of zero indicates that this device has a single port and does not implement the multi-port requirements.

A medium changer (MChngr) bit of one indicates that the device is embedded within or attached to a medium transport element. See SMC for details about medium changers, including a device model for an attached medium changer device. The MChngr bit is valid only when the RMB bit is equal to one. A MChngr bit of zero indicates that the device is not embedded within or attached to a medium transport element.

A relative addressing (RelAdr) bit of one indicates that the device server supports the relative addressing mode. If this bit is set to one, the linked command (Linked) bit shall also be set to one; since relative addressing is only allowed with linked commands. A RelAdr bit of zero indicates the device server does not support relative addressing.

A linked command (Linked) bit of one indicates that the device server supports linked commands (see SAM). A value of zero indicates the device server does not support linked commands.

A command queuing (CmdQue) bit of one indicates that the device supports tagged tasks (command queuing) for this logical unit (see SAM). A value of zero indicates the device server does not support tagged tasks for this logical unit.

ASCII data fields shall contain only graphic codes (i.e., code values 20h through 7Eh). Left-aligned fields shall place any unused bytes at the end of the field (highest offset) and the unused bytes shall be filled with space characters (20h). Right-aligned fields shall place any unused bytes at the start of the field (lowest offset) and the unused bytes shall be filled with space characters (20h).

The vendor identification field contains eight bytes of ASCII data identifying the vendor of the product. The data shall be left aligned within this field.

NOTE 10 It is intended that this field provide a unique vendor identification of the manufacturer of the SCSI device. In the absence of a formal registration procedure, T10 maintains a list of vendor identification codes in use. Vendors are requested to voluntarily submit their identification codes to T10 to prevent duplication of codes (see annex C).

The product identification field contains sixteen bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

The product revision level field contains four bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

7.5.2 SIP-specific INQUIRY data

Portions of bytes 6 and 7 of the standard INQUIRY data shall be used only by the SCSI-3 Interlocked Protocol. These bits are noted in table 19. For details on how the SIP-specific bits relate to the SCSI-3 Interlocked Protocol see SIP. The definitions of the SIP-specific bits shall be as follows.

A ACKQ/REQQ (ACKREQQ) bit of one indicates that the target supports a request and acknowledge data transfer handshake on the secondary bus.

A wide SCSI address 32 (Addr32) bit of one indicates that the target supports 32-bit wide SCSI addresses. A value of zero indicates that the device does not support 32-bit wide SCSI addresses.

A wide SCSI address 16 (Addr16) bit of one indicates that the target supports 16-bit wide SCSI addresses. A value of zero indicates that the device does not support 16-bit wide SCSI addresses.

NOTE 11 If the values of both the Addr16 and Addr32 bits are zero, the device only supports 8-bit wide SCSI addresses.

A wide bus 32 (Wbus32) bit of one indicates that the target supports 32-bit wide data transfers. A value of zero indicates that the device does not support 32-bit wide data transfers.

A wide bus 16 (Wbus16) bit of one indicates that the target supports 16-bit wide data transfers. A value of zero indicates that the device does not support 16-bit wide data transfers.

NOTE 12 If the values of both the Wbus16 and Wbus32 bits are zero, the device only supports 8-bit wide data transfers.

A synchronous transfer (Sync) bit of one indicates that the target supports synchronous data transfer. A value of zero indicates the device does not support synchronous data transfer.

A transfer disable (TranDis) bit of one indicates that the target supports the CONTINUE TASK and TARGET TRANSFER DISABLE messages. A TranDis bit of zero indicates that the device does not support one or both of these messages.

Table 23 defines the relationships between the ACKREQQ, Addr32, Addr16, Wbus32, and Wbus16 bits.

Table 23 – Maximum logical device configuration table

ACKREQQ	Addr32	Addr16	Wbus32	Wbus16	Description
0	0	0	0	0	8 bit wide data path on a single cable with 8 SCSI IDs supported
0	0	0	0	1	16 bit wide data path on a single cable with 8 SCSI IDs supported
0	0	1	0	1	16 bit wide data path on a single cable with 16 SCSI IDs supported
1	0	0	1	0	32 bit wide data path on two cables with 8 SCSI IDs supported
1	0	1	1	0	32 bit wide data path on two cables with 16 SCSI IDs supported
1	1	0	1	0	32 bit wide data path on two cables with 32 SCSI IDs supported

7.5.3 Vital product data

Implementation of vital product data is optional. See 8.4 for details about vital product data. The information returned consists of configuration data (e.g., vendor identification, product identification, model, serial number), manufacturing data (e.g., plant and date of manufacture), field replaceable unit data and other vendor- or device-specific data.

The application client requests the vital product data information by setting the EVPD bit to one and specifying the page code of the desired vital product data (see 8.4). If the device server does not implement the requested page it shall return CHECK CONDITION status. The a sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

NOTES

13 The device server should have the ability to execute the INQUIRY command even when an error occurs that prohibits normal command execution. In such a case, CHECK CONDITION status should be returned for commands other than INQUIRY or REQUEST SENSE. The sense data returned may contain the field replaceable unit code. The vital product data may be obtained for the failing device using the INQUIRY command.

14 This standard defines a format that allows device-independent application client software to display the vital product data returned by the INQUIRY command. The contents of the data may be vendor-specific, and may be unusable without detailed information about the device.

15 This standard does not define the location or method of storing the vital product data. The retrieval of the data may require completion of initialization operations within the device, that may induce delays before the data is available to the application client. Time-critical requirements are an implementation consideration and are not addressed in this standard.

7.5.4 Command support data

Implementation of command support data is optional. The application client may request the command support data information by setting the CmdDt bit to one and specifying the SCSI operation code of the desired CDB.

If the device server implements the requested SCSI operation code, it shall return the data defined in table 24. If the device server does not implement the requested SCSI operation code it shall return the peripheral qualifier and type byte and 001h in the Support field.

Table 24 – Command support data format

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Reserved					Support		
2	ISO version		ECMA version			ANSI-approved version		
3	Reserved							
4	Reserved							
5	CDB size (m - 5)							
6 – m	CDB usage data							

The peripheral qualifier field and the peripheral device type field are defined in 7.5.1.

Table 25 defines the values and meanings of the Support field.

Table 25 – Support values and meanings

Support	Description
000b	Data about the requested SCSI operation code is not currently available.
010b	Reserved
100b	Vendor-specific
110b	Vendor-specific
001b	The device server does not support the tested SCSI operation code. All data after byte 1 is undefined.
011b	The device server supports the tested SCSI operation code in conformance with an SCSI standard. The data format conforms to the definition in table 24.
101b	The device server supports the tested SCSI operation code in a vendor-specific manner. The data format conforms to the definition in table 24.
111b	Reserved

If the Support are 000b, all data after byte 1 is not defined. One possible reason for the Support being 000b is the device server's inability to retrieve information stored on the media. When this is the case, a subsequent request for command support data may be successful.

The ISO, ECMA, and ANSI-approved version fields shall contain standard INQUIRY data naming the standard that defines the SCSI command. Information about standard INQUIRY data may be found in 7.5.1.

The CDB size field shall contain the number of bytes in the CDB for the operation code being queried, and the size of the CDB usage data in the return data.

NOTE 16 The CDB size field is provided primarily for the convenience of the application client. In most cases, the size is known from the operation code group.

The CDB usage data shall contain information about the CDB for the operation code being queried. The first byte of the CDB usage data shall contain the operation code for the operation being queried. All bytes except the first byte of the CDB usage data shall contain a usage map for bits in the CDB for the operation code being queried.

The bits in the usage map shall have a one-for-one correspondence to the CDB for the operation code being queried. If the device server evaluates a bit as all or part of a field in the CDB for the operation code being queried, the usage map shall contain a one in the corresponding bit position. If the device server ignores or treats as reserved a bit in the CDB for the operation code being queried, the usage map shall contain a zero in the corresponding bit position. The usage map bits for a given CDB field all shall have the same value.

Thus, the CDB usage bit map for the INQUIRY command for a device server that implements command support data but not vital product data is: 12h, 02h, FFh, 00h, FFh, 07h. This example assumes that SAM defines uses for only the low-order three bits of the Control byte. Note that the first byte contains the operation code and the remaining bytes contain the usage map.

7.6 LOG SELECT command

The LOG SELECT command (see table 26) provides a means for an application client to manage statistical information maintained by the device about the device or its logical units. Device servers that implement the LOG SELECT command shall also implement the LOG SENSE command. Structures in the form of log parameters within log pages are defined as a way to manage the log data. The LOG SELECT command provides for sending zero or more log pages via the Data-Out Buffer. This standard defines the format of the log pages, but does not define the exact conditions and events that are logged.

Table 26 – LOG SELECT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (4Ch)							
1	Reserved						PCR	SP
2	PC		Reserved					
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Parameter list length							
8	Parameter list length (LSB)							
9	Control							

If reservations are active, they shall affect the execution of the LOG SELECT command as follows. A reservation conflict shall occur when a LOG SELECT command is received from an initiator other than the one holding a logical unit reservation. The LOG SELECT command shall not be affected by extent or element reservations.

A parameter code reset (PCR) bit of one and a parameter list length of zero shall cause all implemented parameters to be set to the target-defined default values (e.g., zero). If the PCR bit is one and the parameter list length is greater than zero, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB. A PCR bit of zero specifies that the log parameters shall not be reset.

A save parameters (SP) bit of one indicates that after performing the specified LOG SELECT operation the target shall save to non-volatile memory all parameters identified as savable by the DS bit in the log page (see 8.2). A SP bit of zero specifies that parameters shall not be saved.

Saving of log parameters is optional and indicated for each log parameter by the DS bit in the page. Log parameters also may be saved at vendor-specific times subject to the TSD bit (see 8.2) in the log parameter or the GLTSD bit in the control mode page (see 8.3.4). If the target does not implement saved parameters for any log parameter and the SP bit is set to one, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

It is not an error to set the SP bit to one and to set the DS bit of a log parameter to one. In this case, the parameter value for that log parameter is not saved.

The page control (PC) field defines the type of parameter values to be selected. The page control field is defined in table 27.

Table 27 − Page control field

Type	LOG SELECT parameter values	LOG SENSE parameter values
00b	Current threshold values	Threshold values
01b	Current cumulative values	Cumulative values
10b	Default threshold values	Default threshold values
11b	Default cumulative values	Default cumulative values

The current cumulative values may be updated by the target or by the application client using the LOG SELECT command to reflect the cumulative number of events experienced by the target. Fields in the parameter control byte (8.2) of each log parameter control the updating and saving of the current cumulative parameters.

The device server shall set the current threshold parameters to the default threshold values in response to a LOG SELECT command with the PC field set to 10b and the parameter list length field set to zero.

The device server shall set all cumulative parameters to their default values in response to a LOG SELECT command with the PC field set to 11b and the parameter list length field set to zero.

The current threshold value may only be modified by the application client via the LOG SELECT command. If the application client attempts to change current threshold values that are not available or not implemented for that log parameter, then the device server shall terminate the LOG SELECT command with CHECK CONDITION status, the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. The saving of current threshold parameters and the criteria for the current threshold being met are controlled by bits in the parameter control byte (8.2).

NOTE 17 Pages or log parameters that are not available may become available at some later time (e.g., after the device has become ready).

The parameter list length field specifies the length in bytes of the parameter list that shall be located in the Data-Out Buffer. A parameter list length of zero indicates that no pages shall be transferred. This condition shall not be considered an error. If an application client sends page codes or parameter codes within the parameter list that are reserved or not implemented by the target, the device server shall terminate the LOG SELECT command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

If a parameter list length results in the truncation of any log parameter, 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 set to INVALID FIELD IN CDB.

The application client should send pages in ascending order by page code value if the Data-Out Buffer contains multiple pages. If the Data-Out Buffer contains multiple log parameters within a page, they should be sent in ascending order by parameter code value. The device server shall return CHECK CONDITION status if the application client sends pages out of order or parameter codes out of order. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

NOTE 18 Initiators should issue LOG SENSE commands prior to issuing LOG SELECT commands to determine supported pages and page lengths.

The target may provide independent sets of log parameters for each logical unit or for each combination of logical units and initiators. If the target does not support independent sets of log parameters and any log parameters are changed that affect other initiators, then the device server shall generate a unit attention condition for all initiators except the one that issued the

LOG SELECT command (see SAM). This unit attention condition shall be returned with an additional sense code of LOG PARAMETERS CHANGED.

If an application client sends a log parameter that is not supported by the target, the device server shall terminate the command with CHECK CONDITION status, set the sense key to ILLEGAL REQUEST, and set the additional sense code to INVALID FIELD IN PARAMETER LIST.

Additional information about the LOG SELECT command may be found in informative annex A.

7.7 LOG SENSE command

The LOG SENSE command (see table 28) provides a means for the application client to retrieve statistical or other operational information maintained by the device about the device or its logical units. It is a complementary command to the LOG SELECT command.

Table 28 – LOG SENSE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (4Dh)							
1	Reserved						PPC	SP
2	PC			Page code				
3	Reserved							
4	Reserved							
5	(MSB) Parameter pointer							
6								(LSB)
7	(MSB) Allocation length							
8								(LSB)
9	Control							

If reservations are active, they shall affect the execution of the LOG SENSE command as follows. A reservation conflict shall occur when a LOG SENSE command is received from an initiator other than the one holding a logical unit reservation. The LOG SENSE command shall not be affected by extent or element reservations.

The parameter pointer control (PPC) bit controls the type of parameters requested from the device server:

a) A PPC bit of one indicates that the device server shall return a log page with parameter code values that have changed since the last LOG SELECT or LOG SENSE command. The device server shall return only those parameter codes following the parameter pointer field.
b) A PPC bit of zero indicates that the log parameter requested from the device server shall begin with the parameter code specified in the parameter pointer field and return the number of bytes specified by the allocation length field in ascending order of parameter codes from the specified log page. A PPC bit of zero and a parameter pointer field of zero shall cause all available log parameters for the specified log page to be returned to the application client subject to the specified allocation length.

Saving parameters is an optional function of the LOG SENSE command. If the target does not implement saving log parameters and if the save parameters (SP) bit is one, then the device server shall return CHECK CONDITION status, set the sense key to ILLEGAL REQUEST, and set the additional sense code to INVALID FIELD IN CDB.

An SP bit of zero indicates the device server shall perform the specified LOG SENSE command and shall not save any log parameters. If saving log parameters is implemented, an SP bit of one indicates that the device server shall perform the specified LOG SENSE command and shall save all log parameters identified as savable by the DS bit (8.2) to a non-volatile, vendor-specific location.

The page control (PC) field defines the type of parameter values to be selected (see 7.6 for the definition of the page control field). The parameter values returned by a LOG SENSE command are determined as follows:

a) The specified parameter values at the last update (in response to a LOG SELECT or LOG SENSE command or done automatically by the target for cumulative values);
b) The saved values, if an update has not occurred since the last power-on or hard reset condition and saved parameters are implemented; or
c) The default values, if an update has not occurred since the last power-on or hard reset condition and saved values are not available or not implemented.

The page code field identifies which page of data is being requested (see 8.2). If the page code is reserved or not implemented, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

The parameter pointer field allows the application client to request parameter data beginning from a specific parameter code to the maximum allocation length or the maximum parameter code supported by the target, whichever is less. If the value of the parameter pointer field is larger than the largest available parameter code known to the device server for the specified page, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

Log parameters within the specified log page shall be transferred in ascending order according to parameter code.

Additional information about the LOG SENSE command may be found in annex A.

7.8 MODE SELECT(6) command

The MODE SELECT(6) command (see table 29) provides a means for the application client to specify medium, logical unit, or peripheral device parameters to the target. Device servers that implement the MODE SELECT command shall also implement the MODE SENSE command. Application clients should issue MODE SENSE prior to each MODE SELECT to determine supported pages, page lengths, and other parameters.

Table 29 – MODE SELECT(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (15h)							
1	Reserved			PF	Reserved			SP
2	Reserved							
3	Reserved							
4	Parameter list length							
5	Control							

If reservations are active, they shall affect the execution of the MODE SELECT command as follows. A reservation conflict shall occur when a MODE SELECT command is received from an initiator other than the one holding a logical unit reservation. If an initiator has an extent or element reservation on an SCSI device, and an another initiator sends a MODE SELECT, a reservation conflict shall occur if the MODE SELECT affects the manner in which access to an extent or element reserved by the first initiator is performed. If the MODE SELECT does not affect access to any reserved extent or element, or mode parameters are saved for each initiator, then a reservation conflict shall not occur.

If a target supports saved pages, it may save only one copy of the page for each logical unit and have it apply to all initiators, or it may save separate copies for each initiator for each logical unit. Multiple port implementations may save one copy per logical unit and have it apply to all initiators on all ports or save a separate copy per logical unit for each initiator on each port. If separate copies are saved, the target shall maintain separate current values for each combination of initiator and logical unit that it detects. Pages that are common to all initiators are not required to have multiple copies.

If an application client sends a MODE SELECT command that changes any parameters applying to other initiators, the device server shall generate a unit attention condition for all initiators except the one that issued the MODE SELECT command (see SAM). The device server shall set the additional sense code to MODE PARAMETERS CHANGED.

The target may provide for independent sets of parameters for each attached logical unit or for each combination of logical unit and initiator. If independent sets of parameters are implemented, and a third party reservation is requested, the device server shall transfer the set of parameters in effect for the application client that sent the RESERVE command to the parameters used for commands from the third party device (see 7.22.3).

A page format (PF) bit of zero indicates that all parameters after the block descriptors are vendor-specific. A PF bit of one indicates that the MODE SELECT parameters following the header and block descriptor(s) are structured as pages of related parameters and are as specified in this standard.

A save pages (SP) bit of zero indicates the device server shall perform the specified MODE SELECT operation, and shall not save any pages. An SP bit of one indicates that the device server shall perform the specified MODE SELECT operation, and shall save to a non-volatile vendor-specific location all the savable pages including any sent in the Data-Out Buffer. The SP bit is optional, even when mode pages are supported by the target. Pages that are saved are identified by the parameter

savable bit that is returned in the page header by the MODE SENSE command (see 8.3). If the PS bit is set in the MODE SENSE data then the page shall be savable by issuing a MODE SELECT command with the SP bit set. If the target does not implement saved pages and the SP bit is set to one, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The parameter list length field specifies the length in bytes of the mode parameter list that shall be contained in the Data-Out Buffer. A parameter list length of zero indicates that the Data-Out Buffer shall be empty. This condition shall not be considered as an error.

The device server shall terminate the command with CHECK CONDITION status if the parameter list length results in the truncation of any mode parameter header, mode parameter block descriptor(s), or mode page. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

The mode parameter list for the MODE SELECT and MODE SENSE commands is defined in 8.3. Parts of each mode parameter list are defined in a device-type dependent manner. Definitions for the parts of each mode parameter list that are uniquely for each device-type may be found in the applicable command standards (see 3.1.11).

The device server shall terminate the MODE SELECT command with CHECK CONDITION status, set the sense key to ILLEGAL REQUEST, set the additional sense code to INVALID FIELD IN PARAMETER LIST, and shall not change any mode parameters for the following conditions:

a) If the application client sets any field that is reported as not changeable by the device server to a value other than its current value;
b) If the application client sets any field in the mode parameter header or block descriptor(s) to an unsupported value;
c) If an application client sends a mode page with a page length not equal to the page length returned by the MODE SENSE command for that page;
d) If the application client sends a unsupported value for a mode parameter and rounding is not implemented for that mode parameter; or
e) If the application client sets any reserved field in the mode parameter list to a non-zero value.

If the application client sends a value for a mode parameter that is outside the range supported by the device server and rounding is implemented for that mode parameter, the device server may either:

a) round the parameter to an acceptable value and terminate the command as described in 5.2; or
b) terminate the command with CHECK CONDITION status, the sense key set to ILLEGAL REQUEST, and set the additional sense code to INVALID FIELD IN PARAMETER LIST.

A device server may alter any mode parameter in any mode page (even those reported as non-changeable) as a result of changes to other mode parameters.

The device server validates the non-changeable mode parameters against the current values that existed for those mode parameters prior to the MODE SELECT command.

NOTE 19 The current values calculated by the device server may affect the application client's operation. The application client may issue a MODE SENSE command after each MODE SELECT command, to determine the current values.

7.9 MODE SELECT(10) command

The MODE SELECT(10) command (see table 30) provides a means for the application client to specify medium, logical unit, or peripheral device parameters to the device server. See the MODE SELECT(6) command (7.8) for a description of the fields and operation of this command. Application clients should issue MODE SENSE prior to each MODE SELECT to determine supported mode pages, mode page lengths, and other parameters. Device servers that implement the MODE SELECT(10) command shall also implement the MODE SENSE(10) command.

Table 30 – MODE SELECT(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (55h)							
1	Reserved			PF	Reserved			SP
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Parameter list length							
8								(LSB)
9	Control							

7.10 MODE SENSE(6) command

The MODE SENSE(6) command (see table 31) provides a means for a device server to report parameters to an application client. It is a complementary command to the MODE SELECT(6) command.

Table 31 – MODE SENSE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (1Ah)							
1	Reserved				DBD	Reserved		
2	PC		Page code					
3	Reserved							
4	Allocation length							
5	Control							

If reservations are active, they shall affect the execution of the MODE SENSE command as follows. A reservation conflict shall occur when a MODE SENSE command is received from an initiator other than the one holding a logical unit reservation. The MODE SENSE command shall not be affected by extent or element reservations.

A disable block descriptors (DBD) bit of zero indicates that the device server may return zero or more block descriptors in the returned MODE SENSE data (see 8.3), at the device server's discretion. A DBD bit of one specifies that the device server shall not return any block descriptors in the returned MODE SENSE data.

The page control (PC) field defines the type of mode parameter values to be returned in the mode pages. The page control field is defined in table 32.

Table 32 – Page control field

Code	Type of parameter	Clause
00b	Current values	7.10.1
01b	Changeable values	7.10.2
10b	Default values	7.10.3
11b	Saved values	7.10.4

NOTE 20 The page control field only affects the mode parameters within the mode pages, however the PS bit, page code and page length fields should return current values since they have no meaning when used with other types. The mode parameter header and mode parameter block descriptor should return current values.

The page code specifies which mode page(s) to return. Mode page code usage is defined in table 33.

Table 33 – Mode page code usage for all devices

Page code	Description
00h	Vendor-specific (does not require page format)
01h - 1Fh	See specific device-types
20h - 3Eh	Vendor-specific (page format required)
3Fh	Return all mode pages

An application client may request any one or all of the supported mode pages from the device server. If an application client issues a MODE SENSE command with a page code value not implemented by the target, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN CDB.

A page code of 3Fh indicates that all mode pages implemented by the target shall be returned to the application client. If the mode parameter list exceeds 256 bytes for a MODE SENSE(6) command or 65 536 bytes for a MODE SENSE(10) command, 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.

Mode page 00h, if implemented, shall be returned after all other mode pages.

NOTES
21 Mode pages should be returned in ascending page code order except for mode page 00h.
22 If the PC field and the page code field are both set to zero the device server should return a mode parameter header and block descriptor (if applicable).

The mode parameter list for all device types for MODE SELECT and MODE SENSE is defined in 8.3. Parts of the mode parameter list are specifically defined for each device type. Definitions for the parts of each mode parameter list that are unique for each device-type may be found in the applicable command standards (see 3.1.11).

7.10.1 Current values

A PC field value of 00b requests that the device server return the current values of the mode parameters. The current values returned are:

a) the current values of the mode parameters established by the last successful MODE SELECT command;
b) the saved values of the mode parameters if a MODE SELECT command has not successfully completed since the last power-on or hard reset condition; or
c) the default values of the mode parameters, if saved values, are not available or not supported.

7.10.2 Changeable values

A PC field value of 01b requests that the device server return a mask denoting those mode parameters that are changeable. In the mask, the fields of the mode parameters that are changeable shall be set to all one bits and the fields of the mode parameters that are non-changeable (i.e., defined by the target) shall be set to all zero bits.

NOTES
23 An attempt to change a non-changeable mode parameter (via MODE SELECT) results in an error condition (see 7.8).
24 The application client should issue a MODE SENSE command with the PC field set to 01b and the page code field set to 3Fh to determine which mode pages are supported, which mode parameters within the mode pages are changeable, and the supported length of each mode page prior to issuing any MODE SELECT commands.

7.10.3 Default values

A PC field value of 10b requests that the device server return the default values of the mode parameters. Unsupported parameters shall be set to zero. Default values should be accessible even if the device is not ready.

7.10.4 Saved values

A PC field value of 11b requests that the device server return the saved values of the mode parameters. Implementation of saved page parameters is optional. Mode parameters not supported by the target shall be set to zero. If saved values are not implemented, the command shall be terminated with CHECK CONDITION status, the sense key set to ILLEGAL REQUEST and the additional sense code set to SAVING PARAMETERS NOT SUPPORTED.

NOTE 25 The method of saving parameters is vendor-specific. The parameters are preserved in such a manner that they are retained when the device is powered down. All savable pages should be considered saved when a MODE SELECT command issued with the SP bit set to one has returned a GOOD status or after the successful completion of a FORMAT UNIT command.

7.10.5 Initial responses

After a power-up condition or hard reset condition, the device server shall respond in the following manner:

a) If default values are requested, report the default values;
b) If saved values are requested, report valid restored mode parameters, or restore the mode parameters and report them. If the saved values of the mode parameters are not able to be accessed from the non-volatile vendor-specific location, terminate the command with CHECK CONDITION status and set the sense key set to NOT READY. If saved parameters are not implemented respond as defined in 7.10.4; or

c) If current values are requested and the current values of the mode parameters have not been sent by the application client (via a MODE SELECT command), the device server may return either the default or saved values, as defined above. If current values have been sent, the current values shall be reported.

7.11 MODE SENSE(10) command

The MODE SENSE(10) command (see table 34) provides a means for a device server to report parameters to an application client. It is a complementary command to the MODE SELECT(10) command. Device servers that implement the MODE SENSE(10) command shall also implement the MODE SELECT(10) command. See the MODE SENSE(6) command (7.10) for a description of the fields and operation of this command.

Table 34 — MODE SENSE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (5Ah)							
1	Reserved				DBD	Reserved		
2	PC		Page code					
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)							
8	Allocation length							(LSB)
9	Control							

7.12 PERSISTENT RESERVE IN command

The PERSISTENT RESERVE IN command (see table 35) is used to obtain information about persistent reservations and reservation keys that are active within a device server. This command is used in conjunction with the PERSISTENT RESERVE OUT command.

Table 35 – PERSISTENT RESERVE IN command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (5Eh)							
1	Reserved			Service action				
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)			Allocation length				
8								(LSB)
9	Control							

When a device server receives a PERSISTENT RESERVE IN command and RESERVE(6) or RESERVE(10) logical unit or extent reservations or SMC element reservations are active (see 7.22), the command shall be rejected with a RESERVATION CONFLICT status.

The actual length of the PERSISTENT RESERVE IN parameter data is available in a parameter data field. The Allocation length field in the CDB indicates how much space has been reserved for the returned parameter list. If the length is not sufficient to contain the entire parameter list, the first portion of the list shall be returned. This shall not be considered an error. If the remainder of the list is required, the application client should send a new PERSISTENT RESERVE IN command with a Allocation length field large enough to contain the entire list.

7.12.1 PERSISTENT RESERVE IN Service Actions

Service actions that require access to the persistent reservation and registration information may require the enabling of a nonvolatile memory within the logical unit. If the nonvolatile memory is not ready, the device server shall return CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense data shall be set as described in the TEST UNIT READY command (see 7.24).

The Service action codes for the PERSISTENT RESERVE IN command are defined in table 36.

Table 36 – PERSISTENT RESERVE IN Service Action Codes

Code	Name	Description
00h	Read Keys	Reads all registered Reservation Keys
01h	Read Reservations	Reads all current persistent reservations
02-1Fh	Reserved	Reserved

7.12.1.1 Read Keys

The Read Keys service action requests that the device server return a parameter list containing a header and a complete list of all reservation keys currently registered with the device server. The keys may have been passed by a PERSISTENT RESERVE OUT command that has performed a Register service action. The relationship between a reservation key and the initiator or port is outside the scope of this standard.

7.12.1.2 Read Reservations

The Read Reservations service action requests that the device server return a parameter list containing a header and a complete list of all persistent reservations that are presently active in the device server and its extents.

7.12.2 PERSISTENT RESERVE IN parameter data for Read Keys

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the Read Keys service action is shown in table 37.

Table 37 — PERSISTENT RESERVE IN parameter data for Read Keys

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				Generation				
3								(LSB)
4	(MSB)							
				Additional length (n-7)				
7								(LSB)
				Reservation key list				
8	(MSB)							
				First reservation key				
15								(LSB)
				.				
				.				
				.				
n-7	(MSB)							
				Last reservation key				
n								(LSB)

The Generation value is a 32-bit counter in the device server that shall be incremented every time a PERSISTENT RESERVE OUT command requests a Register, a Clear, a Preempt, or a Preempt and Clear operation. The counter shall not be incremented by a PERSISTENT RESERVE IN command, by a PERSISTENT RESERVE OUT command that performs a Reserve or Release service action, or by a PERSISTENT RESERVE OUT command that is not performed due to an error or reservation conflict. The Generation value shall be set to 0 as part of the power on reset process.

The Generation value allows the application client examining the generation value to verify that the configuration of the initiators attached to a logical unit has not been modified by another application client without the knowledge of the examining application client.

The Additional length field contains a count of the number of bytes in the Reservation key list. If the Allocation length specified by the PERSISTENT RESERVE IN command is not sufficient to contain the entire parameter list, then only the

bytes from 0 to the maximum allowed Allocation length shall be sent to the application client. The remaining bytes shall be truncated, although the Additional length field shall still contain the actual number of bytes in the reservation key list without consideration of any truncation resulting from an insufficient Allocation length. This shall not be considered an error.

The Reservation key list contains all the 8-byte reservation keys registered with the device server through PERSISTENT RESERVE OUT Reserve, Preempt, Preempt and Clear, or Register service actions. Each reservation key may be examined by the application client and correlated with a particular initiator and SCSI port by mechanisms outside the scope of this standard.

7.12.3 PERSISTENT RESERVE IN parameter data for Read Reservations

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the Read Reservations service action is shown in table 38.

Table 38 — PERSISTENT RESERVE IN parameter data for Read Reservations

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				Generation				
3								(LSB)
4	(MSB)							
				Additional length (n-7)				
7								(LSB)
8								
				Reservation descriptors				
n				(see table 39)				(LSB)

The Generation field shall be as defined for the PERSISTENT RESERVE IN Read Keys parameter data.

The Additional length field contains a count of the number of bytes in of Reservation descriptors. If the Allocation length specified by the PERSISTENT RESERVE IN command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed Allocation length shall be sent to the application client. The remaining bytes shall be truncated, although the Additional length field shall still contain the actual number of bytes of Reservation descriptors and shall not be affected by the truncation. This shall not be considered an error.

The format of a single read Reservation descriptor is defined in table 39. There shall be one read Reservation descriptor for each persistent reservation held on the logical unit by any initiator.

Table 39 – PERSISTENT RESERVE IN Read Reservation Descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7	Reservation key							(LSB)
8	(MSB)							
11	Scope-specific address							(LSB)
12	Reserved							
13	Scope				Type			
14	(MSB)							
15	Extent length							(LSB)

For each persistent reservation held on the logical unit, there shall be a read Reservation descriptor presented in the list of parameter data returned by the device server in response to the PERSISTENT RESERVE IN command with a Read Reservations action. The descriptor shall contain the Reservation Key under which the persistent reservation is held. The Type and Scope of the persistent reservation as present in the PERSISTENT RESERVE OUT command that created the persistent reservation shall be returned (see 7.12.3.1 and 7.12.3.2).

Reservation key is the registered reservation key under which the reservation is held. Using techniques that are outside the scope of this standard, the application should be able to associate the reservation key with the initiator that holds the reservation.

If the Scope is an Extent reservation, the Scope-specific address field shall contain the LBA of the first block of the extent and the Extent length field shall contain the number of blocks in the extent. If the Scope is an Element reservation, the Scope-specific address field shall contain the Element address, zero filled in the most significant bytes to fit the field, and the Extent length field shall be set to zero. If the Scope is a Logical Unit reservation, both the Scope-specific address and Extent length fields shall be set to zero.

7.12.3.1 Persistent Reservations Scope

The value in the Scope field shall indicate whether a persistent reservation applies to an entire logical unit, to a portion of the logical unit defined as an extent, or to an element. The values in the Scope field are defined in table 40.

Table 40 – Persistent Reservation Scope Codes

Code	Name	Description
0h	LU	Persistent reservation applies to the full logical unit
1h	Extent	Persistent reservation applies to the specified extent
2h	Element	Persistent reservation applies to the specified element
3-Fh	Reserved	Reserved

7.12.3.1.1 LU Scope

A Scope field value of LU shall indicate that the persistent reservation applies to the entire logical unit. The LU scope shall be implemented by all device servers that implement PERSISTENT RESERVE OUT.

7.12.3.1.2 Extent Scope

A Scope field value of Extent shall indicate that the persistent reservation applies to the extent of the logical unit defined by the Scope-specific address and Extent length fields in the PERSISTENT RESERVE OUT parameter list. An extent is defined only for devices defining contiguous logical block addresses. The Extent scope is optional for all device servers that implement PERSISTENT RESERVE OUT. The number of extents that may be reserved for a logical unit is vendor-specific.

7.12.3.1.3 Element Scope

A Scope field value of Element shall indicate that the persistent reservation applies to the element of the logical unit defined by the Scope-specific address field in the PERSISTENT RESERVE OUT parameter list. An element is defined by the SCSI-3 Medium Changer Commands (SMC) standard. The Element scope is optional for all device servers that implement PERSISTENT RESERVE OUT.

7.12.3.2 Persistent Reservations Type

The value in the Type field shall specify the characteristics of the persistent reservation being established for all data blocks within the extent or within the logical unit. Table 41 defines the characteristics of the five different type values. For each persistent reservation type, table 41 lists code value and describes the required device server support. In table 41, the description of required device server support is divided into three paragraphs. The first paragraph defines the required handling for read operations. The second paragraph defines the required handling for write operations. The third paragraph defines the handling for subsequent attempts to establish persistent reservations.

Table 41 – Persistent Reservation Type Codes

Code	Name	Description
0h	Read Shared	**Reads Shared**: Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator. **Writes Prohibited**: Any command from any initiator that performs a transfer from the initiator to the storage medium or cache of the logical unit shall result in a reservation conflict. **Additional Reservations Allowed**: Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 42.

(continued)

Table 41 — Persistent Reservation Type Codes (continued)

Code	Name	Description
1h	Write Exclusive	**Reads Shared**: Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator. **Writes Exclusive**: Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the initiator to the storage medium or cache of the logical unit shall result in a reservation conflict. **Additional Reservations Allowed**: Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 42.
2h	Read Exclusive	**Reads Exclusive**: Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the storage medium or cache of the logical unit to the initiator shall result in a reservation conflict. **Writes Shared**: Any application client on any any initiator may execute commands that perform transfers from the initiator to the storage medium or cache of the logical unit. **Additional Reservations Allowed**: Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 42.
3h	Exclusive Access	**Reads Exclusive**: Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the storage medium or cache of the logical unit to the initiator shall result in a reservation conflict. **Writes Exclusive**: Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the initiator to the storage medium or cache of the logical unit shall result in a reservation conflict. **Additional Reservations Restricted**: Any Persistent Reserve Out command with the Reserve service action from any initiator other than the initiator holding the persistent reservation shall result in a reservation conflict. The initiator that holds the persistent reservation may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 42.

(continued)

Table 41 – Persistent Reservation Type Codes (concluded)

Code	Name	Description
4h	Shared Access	**Reads Shared**: Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator. **Writes Shared**: Any application client on any any initiator may execute commands that perform transfers from the initiator to the storage medium or cache of the logical unit. **Additional Reservations Restricted**: Any Persistent Reserve Out command with the Reserve service action from any initiator other than the initiator holding the persistent reservation shall result in a reservation conflict. The initiator that holds the persistent reservation may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 42.
5-Fh	Reserved	

Table 42 – New Persistent Reservation Conflicts With Existing

Persistent Reservation That Is Being Attempted		**Persistent Reservation That Is Held**									
		Read Shared		Write Exclusive		Read Exclusive		Exclusive Access*		Shared Access*	
		LU	EX	LU	EX	LU	EX	LU	EX	LU	EX
Read Shared	LU	N	N	Y	Y	Y	Y	Y	Y	N	N
	EX	N	N	Y	O	Y	O	Y	O	N	N
Read Exclusive	LU	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	EX	Y	O	Y	O	Y	O	Y	O	Y	O
Write Exclusive	LU	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	EX	Y	O	Y	O	Y	O	Y	O	Y	O
Exclusive Access*	LU	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	EX	Y	O	Y	O	Y	O	Y	O	Y	O
Shared Access*	LU	N	N	Y	Y	Y	Y	Y	Y	N	N
	EX	N	N	Y	O	Y	O	Y	O	N	N

Key:
LU = Logical Unit scope
EX = Extent or Element scope
* = Conflicts with all reservation requests from other initiators

N = no conflict
Y = conflict
O = conflict occurs if extent or element overlaps with existing extent or element reservation

7.13 PERSISTENT RESERVE OUT command

The PERSISTENT RESERVE OUT command (see table 43) is used to reserve a logical unit or an extent within a logical unit for the exclusive or shared use of a particular initiator. The command shall be used in conjunction with the PERSISTENT RESERVE IN command and shall not be used with the RESERVE and RELEASE commands.

Persistent reservations shall conflict with reservations established by the RESERVE command. Initiators performing PERSISTENT RESERVE OUT Service actions are identified by a reservation key provided by the application client. An application client may use the PERSISTENT RESERVE IN command to identify which initiators are holding conflicting or invalid persistent reservations and use the PERSISTENT RESERVE OUT command to preempt those reservations if required.

Since persistent reservations are not reset by the TARGET RESET task management function or other global actions, they may be used to enforce device sharing among multiple initiators. The PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands provide the basic mechanism for dynamic contention resolution in multiple-initiator systems using multiple port targets. The identification of persistent reservations using the reservation key makes it possible to determine which ports hold conflicting persistent reservations and to take over persistent reservations from failing or uncooperative initiators.

Table 43 – PERSISTENT RESERVE OUT command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (5Fh)							
1	Reserved				Service action			
2	Scope				Type			
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)							
8	Parameter list length (18h)							(LSB)
9	Control							

When a device server receives a PERSISTENT RESERVE OUT command and RESERVE(6) or RESERVE(10) logical unit or extent reservations or SMC element reservations are active (see 7.22), the command shall be rejected with a RESERVATION CONFLICT status.

Commands from any initiator that conflict with a successfully established persistent reservation shall be rejected with a status of RESERVATION CONFLICT. The following commands shall not conflict with a reservation established by the PERSISTENT RESERVE OUT command:

PERSISTENT RESERVE IN
PERSISTENT RESERVE OUT (with an Service action of Preempt)
PERSISTENT RESERVE OUT (with an Service action of Preempt and Clear)
PERSISTENT RESERVE OUT (with a Reserve service action that does not conflict with established persistent reservations or tasks)

The PERSISTENT RESERVE OUT command contains fields that specify a persistent reservation Service action, the intended scope of the persistent reservation, and the restrictions caused by the persistent reservation. The Type and Scope fields are defined in 7.12.3.1 and 7.12.3.2. If a Scope field specifies a scope that is not implemented, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID FIELD IN CDB.

Fields contained in the PERSISTENT RESERVE OUT parameter list specify the reservation keys and extent information required to perform a particular persistent reservation Service action.

The parameter list shall be 24 bytes in length and the Parameter list length field shall contain 24. If the Parameter list length is not 24, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to PARAMETER LIST LENGTH ERROR.

The capability of preserving persistent reservations and registration keys requires the use of a nonvolatile memory within the logical unit. If the nonvolatile memory is not accessible at the time that a PERSISTENT RESERVE OUT command attempts to activate the Persist Through Power Loss capability, the device server shall return CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense data shall be set as described in the TEST UNIT READY command (see 7.24).

7.13.1 PERSISTENT RESERVE OUT Service Actions

When processing the PERSISTENT RESERVE OUT service actions, the device server shall increment the generation value as specified in 7.12.2.

The PERSISTENT RESERVE OUT command Service actions are defined in table 44.

Table 44 – PERSISTENT RESERVE OUT Service Action Codes

Code	Name	Description
00h	Register	Register a reservation key with the device server
01h	Reserve	Create a persistent reservation using a reservation key
02h	Release	Release a persistent reservation
03h	Clear	Clear all reservation keys and all persistent reservations
04h	Preempt	Preempt persistent reservations from another initiator
05h	Preempt & clear	Preempt persistent reservations from another initiator and clear the task set for the preempted initiator
06-1Fh	Reserved	

7.13.1.1 Register

The PERSISTENT RESERVE OUT command executing a Register service action registers a reservation key with a device server without generating a reservation. For each initiator that performs a PERSISTENT RESERVE OUT Register service action, the device server shall retain the reservation key until the key is changed by a new PERSISTENT RESERVE OUT command with the Register service action from the same initiator or until the key is reset to the default value of zero by powering down the logical unit, if the last APTPL received by the device server was zero (see 7.13.2) or by performing a Clear, Preempt, or Preempt and Clear service action.

The Register service action may be performed regardless of any active persistent reservations. All existing persistent reservations for the initiator receive the new reservation key.

7.13.1.2 Reserve

The PERSISTENT RESERVE OUT command performing a Reserve service action creates a persistent reservation having a specified scope and type. The scope and type of a persistent reservation are defined in 7.12.3.1 and 7.12.3.2.

A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Reserve service action that conflicts with any active persistent reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT is enabled for execution. The PERSISTENT RESERVE OUT command with a Reserve service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.

NOTE 26 For the simplest predictable behavior, the Reserve service action should be performed with the Ordered task attribute.

Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt or Preempt and Clear service action. New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The persistent reservation of a logical unit or the persistent reservation of extents having the same type value shall be permitted if no conflicting persistent reservations are held by another initiator. When such overlapping persistent reservations are released, each of the extent reservations and the logical unit reservation shall be removed with a separate Release service action.

A persistent reservation shall be tested for conflicts with other persistent reservations as described above and shall take effect when the task executing the PERSISTENT RESERVE OUT command enters the enabled task state.

7.13.1.3 Release

The PERSISTENT RESERVE OUT command performing a Release service action removes an active persistent reservation held by the same initiator. The fields associated with the Release service action shall match fields of the active persistent reservation. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Release service action when no persistent reservation exists from that initiator. The reservation key shall not be changed by the Release service action.

The device server shall return a CHECK CONDITION status for a PERSISTENT RESERVE OUT command that specifies the release of a persistent reservation matching some but not all of the scope, reservation key, and extent values. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID RELEASE OF ACTIVE PERSISTENT RESERVATION. Attempts to release persistent reservations where of the scope, reservation key, and extent values match an existing persistent reservation shall not be considered errors.

An active persistent reservation may also be released by either of the following mechanisms:

a) Power off. When the most recent APTPL value received by the device server is zero, a power off performs a hard reset, clears all persistent reservations, and sets reservation keys to their default value of zero (see 7.13.2); or
b) Execution of a PERSISTENT RESERVE OUT command from another initiator with a Persistent Reserve service action of Preempt or Preempt and Clear.

The PERSISTENT RESERVE OUT command with a Release service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.

A Release service action should not be performed if any operations interlocked by the persistent reservation are not yet complete.

7.13.1.4 Clear

The PERSISTENT RESERVE OUT command that successfully performs a Clear service action shall remove all persistent reservations for all initiators. All reservation keys shall be reset to default value of zero. Any commands from any initiator that have been accepted by the device server as nonconflicting shall continue normal execution.

A Unit Attention condition shall be established for all initiators for the cleared logical unit. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED.

The PERSISTENT RESERVE OUT command with a Clear service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.

The Clear service action should not be performed except during recoveries that are associated with initiator or system reconfiguration, since data integrity may be compromised.

7.13.1.5 Preempt

The PERSISTENT RESERVE OUT command that successfully performs a Preempt service action shall remove all persistent reservations for the initiator specified by the PERSISTENT RESERVE OUT parameter list. The initiator is identified by the reservation key of the initiator to be preempted. Any commands from any initiator that have been accepted by the device server as nonconflicting shall continue normal execution.

A Unit Attention condition is established for the preempted initiator. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. Subsequent commands are subject to the persistent reservation restrictions established by the preempting initiator.

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PERSISTENT RESERVE OUT command and the corresponding fields in the PERSISTENT RESERVE OUT parameter list.

The registration key for the initiator that has been preempted shall be reset to default value of zero by the Preempt service action.

A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Preempt service action that conflicts with any active persistent reservations except the preempted reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT is enabled for execution. The PERSISTENT RESERVE OUT command with a Preempt service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.

NOTE 27 For the simplest predictable behavior, the Preempt service action should be performed with the Ordered task attribute.

Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt or Preempt and Clear service action. New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The persistent reservation of a logical unit or the persistent reservation of extents having the same type value shall be permitted if no conflicting persistent reservations other than the reservations being preempted are held by another initiator.

7.13.1.6 Preempt and Clear

The PERSISTENT RESERVE OUT command performing a Preempt and Clear service action removes all persistent reservations for the initiator specified by the PERSISTENT RESERVE OUT parameter list. The initiator is identified by the reservation key of the initiator to be preempted. Any commands from the initiator being preempted are each terminated as if an ABORT TASK task management function had been performed by the preempted initiator.

A Unit Attention condition is established for the preempted initiator. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. Subsequent new commands and retries of commands that timed out because they were cleared are subject to the persistent reservation restrictions established by the preempting initiator.

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PERSISTENT RESERVE OUT command and the corresponding fields in the PERSISTENT RESERVE OUT parameter list.

The Preempt and Clear service action shall clear any ACA condition associated with the initiator being preempted and shall clear any tasks with an ACA attribute from that initiator. ACA conditions for other initiators shall prevent the execution of the PERSISTENT RESERVE OUT task, which shall end with status of ACA ACTIVE.

NOTE 28 The Preempt and Clear service action will clear the ACA condition associated with the initiator being preempted eventhough the task is terminated with an ACA ACTIVE status. Thus, the next command arriving at the device server will not encounter the ACA condition previously active for the initiator being preempted.

Any Asynchronous Event Reporting operations in progress that were initiated by the device server are not affected by the Preempt and Clear service action.

The reservation key registered for the initiator that has been preempted shall be reset to the default value of zero by the Preempt and Clear service action.

A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Preempt and Clear service action that conflicts with any active persistent reservations except the preempted reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT is enabled for execution. The PERSISTENT RESERVE OUT command with a Preempt service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.

NOTE 29 For the simplest predictable behavior, the Preempt and Clear service action should be performed with the Ordered task attribute.

Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt or Preempt and Clear service action. New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The persistent reservation of a logical unit or the persistent reservation of extents having the same type value shall be permitted if no conflicting persistent reservations other than the reservations being preempted are held by another initiator.

7.13.2 PERSISTENT RESERVE OUT parameter list

The parameter list required to perform the PERSISTENT RESERVE OUT command are defined in table 45. All fields shall be sent on all PERSISTENT RESERVE OUT commands, even if the field is not required for the specified Service action and Scope values.

Table 45 − PERSISTENT RESERVE OUT parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7				Reservation key				(LSB)
8	(MSB)							
15				Service Action Reservation key				(LSB)
16	(MSB)							
19				Scope-specific address				(LSB)
20				Reserved				APTPL
21				Reserved				
22	(MSB)							
23				Extent length				(LSB)

The Reservation key field contains an 8-byte token provided by the application client to the device server to identify the initiator that is the source of the PERSISTENT RESERVE OUT command. The device server shall verify that the Reservation key field in a PERSISTENT RESERVE OUT command matches the registered reservation key for the initiator from which the command was received. If a PERSISTENT RESERVE OUT command specifies a Reservation key field other than the reservation key registered for the initiator, the device server shall return a RESERVATION CONFLICT status. The reservation key of the initiator shall be valid for all Service action and Scope values.

The Service Action Reservation key field contains information needed for three service actions; the Register, Preempt, and Preempt and Clear service actions. For the Register service action, the Service Action Reservation key field contains the new

reservation key to be registered. For the Preempt and Preempt and Clear service actions, the Service Action Reservation key field contains the reservation key of the persistent reservation that is being preempted. For the Preempt and Preempt and Clear service actions, failure of the Service Action Reservation key to match any registered reservation keys shall result in the device server returning a RESERVATION CONFLICT status. The Service Action Reservation key is ignored for all service actions except those described in this paragraph.

If the Scope is an Extent reservation, the Scope-specific address field shall contain the LBA of the first block of the extent and the Extent length field shall contain the number of blocks in the extent. If the Scope is an Element reservation, the Scope-specific address field shall contain the Element address, zero filled in the most significant bytes to fit the field, and the Extent length field shall be set to zero. If the Service action is Register or Clear or if the Scope is a Logical Unit reservation, both the Scope-specific address and Extent length fields shall be set to zero.

The Activate Persist Through Power Loss (APTPL) bit shall be valid only for the Register service action. In all other cases, the APTPL shall be ignored. Support for an APTPL bit equal to one is optional. If a device server that does not support the APTPL bit value of one receives that value in a Register service action, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID FIELD IN PARAMETER LIST.

If the last valid APTPL bit value received by the device server is zero, the loss of power in the target shall release all persistent reservations and set all reservation keys to their default value of zero. If the last valid APTPL bit value received by the device server is one, the logical unit shall retain all persistent reservations and all reservation keys for all initiators even if power is lost and later returned. The most recently received valid APTPL value from any initiator shall govern logical unit's behavior in the event of power loss.

Table 46 summarizes which fields are set by the application client and interpreted by the device server for each Service action and Scope value. Two PERSISTENT RESERVE OUT parameters are not summarized in table 46; Reservation key and APTPL.

Table 46 – PERSISTENT RESERVE OUT Service actions and valid parameters

		Parameters	
Service action	**Allowed Scope**	Service Action Reservation key	Extent or Element Parameters
Register	LU	valid	ignored
Reserve Reserve Reserve	LU Extent Element	ignored ignored ignored	ignored Extent valid Element valid
Release Release Release	LU Extent Element	ignored ignored ignored	ignored Extent valid Element valid
Clear	LU	ignored	ignored
Preempt Preempt Preempt	LU Extent Element	valid valid valid	ignored Extent valid Element valid
Preempt & clear Preempt & clear Preempt & clear	LU Extent Element	valid valid valid	ignored Extent valid Element valid

7.14 PREVENT ALLOW MEDIUM REMOVAL command

The PREVENT ALLOW MEDIUM REMOVAL command (see table 47) requests that the target enable or disable the removal of the medium in the logical unit. The logical unit shall not allow medium removal if any initiator currently has medium removal prevented.

Table 47 – 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							

If reservations are active, they shall affect the execution of the PREVENT ALLOW MEDIUM REMOVAL command as follows. Receipt of a PREVENT ALLOW MEDIUM REMOVAL command with a Prevent value of zero shall not cause a reservation conflict under any circumstances. A reservation conflict shall occur when a PREVENT ALLOW MEDIUM REMOVAL command with a non-zero Prevent value is received from an initiator other than the one holding a reservation.

Table 48 defines the Prevent values and their meanings.

Table 48 – PREVENT ALLOW MEDIUM REMOVAL Prevent field

Prevent	Description
00b	Medium removal shall be allowed from both the data transport element and the attached medium changer (if any).
01b	Medium removal shall be prohibited from the data transport element but allowed from the attached medium changer (if any).
10b	Medium removal shall be allowed for the data transport element but prohibited for the attached medium changer.
11b	Medium removal shall be prohibited for both the data transport element and the attached medium changer.

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

The prevention of medium removal shall begin when any application client issues a PREVENT ALLOW MEDIUM REMOVAL command with a Prevent bit of one (medium removal prevented). The prevention of medium removal for the logical unit shall terminate:

a) after all initiators with application clients that previously prevented medium removal issue PREVENT ALLOW MEDIUM REMOVAL commands with a Prevent bit of zero, and the device server has successfully performed a synchronize cache operation; or
b) upon a hard reset condition.

While a prevention of medium removal condition is in effect, the target shall inhibit mechanisms that normally allow removal of the medium by an operator.

7.15 READ BUFFER command

The READ BUFFER command (see table 49) is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing memory in the SCSI device and the integrity of the service delivery subsystem. This command shall not alter the medium.

Table 49 — READ BUFFER command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (3Ch)							
1	Reserved					Mode		
2	Buffer ID							
3	(MSB)							
4	Buffer offset							
5								(LSB)
6	(MSB)							
7	Allocation length							
8								(LSB)
9	Control							

If reservations are active, they shall affect the execution of the READ BUFFER command as follows. A reservation conflict shall occur when a READ BUFFER command is received from an initiator other than the one holding a logical unit reservation. The READ BUFFER command shall not be affected by extent or element reservations.

The function of this command and the meaning of fields within the command descriptor block depend on the contents of the mode field. The mode field is defined in table 50.

Table 50 − READ BUFFER mode field

Mode	Description	Type
000b	Combined header and data	Optional
001b	Vendor-specific	Vendor-specific
010b	Data	Optional
011b	Descriptor	Optional
100b	Reserved	Reserved
101b	Reserved	Reserved
110b	Reserved	Reserved
111b	Reserved	Reserved

7.15.1 Combined header and data mode (000b)

In this mode, a four-byte header followed by data bytes is returned to the application client in the Data-In Buffer. The buffer ID and the buffer offset fields are reserved.

The four-byte READ BUFFER header (see table 51) is followed by data bytes from the buffer.

Table 51 − READ BUFFER header

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	(MSB)							
				Buffer capacity				
3								(LSB)

The buffer capacity field specifies the total number of data bytes available in the buffer. This number is not reduced to reflect the allocation length; nor is it reduced to reflect the actual number of bytes written using the WRITE BUFFER command. Following the READ BUFFER header, the device server shall transfer data from the buffer. The device server shall terminate filling the Data-In Buffer when allocation length bytes of header plus data have been transferred or when all available header and buffer data have been transferred to the application client, whichever is less.

7.15.2 Vendor-specific mode (001b)

In this mode, the meanings of the buffer ID, buffer offset, and allocation length fields are not specified by this standard.

7.15.3 Data mode (010b)

In this mode, the Data-In Buffer is filled only with logical unit buffer data. The buffer ID field identifies a specific buffer within the logical unit from which data shall be transferred. The vendor assigns buffer ID codes to buffers within the logical unit. Buffer ID zero shall be supported. If more than one buffer is supported, additional buffer ID codes shall be assigned contiguously, beginning with one. Buffer ID code assignments for the READ BUFFER command shall be the same as for the WRITE BUFFER command. If an unsupported buffer ID code is selected, the device server shall return CHECK CONDITION status, shall set the sense key to ILLEGAL REQUEST, and set the additional sense code to ILLEGAL FIELD IN CDB.

The device server shall terminate filling the Data-In Buffer when allocation length bytes have been transferred or when all the available data from the buffer has been transferred to the application client, whichever amount is less.

The buffer offset field contains the byte offset within the specified buffer from which data shall be transferred. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor (see 7.15.4). If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status, shall set the sense key to ILLEGAL REQUEST, and set the additional sense code to ILLEGAL FIELD IN CDB.

7.15.4 Descriptor mode (011b)

In this mode, a maximum of four bytes of READ BUFFER descriptor information is returned. The device server shall return the descriptor information for the buffer specified by the buffer ID (see the description of the buffer ID in 7.15.3). If there is no buffer associated with the specified buffer ID, the device server shall return all zeros in the READ BUFFER descriptor. The buffer offset field is reserved in this mode. The allocation length should be set to four or greater. The device server shall transfer the lesser of the allocation length or four bytes of READ BUFFER descriptor. The READ BUFFER descriptor is defined as shown in table 52.

Table 52 − READ BUFFER descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	Offset boundary							
1	(MSB)							
				Buffer capacity				
3								(LSB)

The offset boundary field returns the boundary alignment within the selected buffer for subsequent WRITE BUFFER and READ BUFFER commands. The value contained in the offset boundary field shall be interpreted as a power of two.

The value contained in the buffer offset field of subsequent WRITE BUFFER and READ BUFFER commands should be a multiple of 2**(offset boundary) as shown in table 53.

Table 53 − Buffer offset boundary

Offset boundary	2**Offset boundary	Buffer offsets
0	2**0 = 1	Byte boundaries
1	2**1 = 2	Even-byte boundaries
2	2**2 = 4	Four-byte boundaries
3	2**3 = 8	Eight-byte boundaries
4	2**4 = 16	16-byte boundaries
.	.	.
FFh	Not applicable	0 is the only supported buffer offset.

The buffer capacity field shall return the size of the selected buffer in bytes.

NOTE 30 In a system employing multiple application clients, a buffer may be altered between the WRITE BUFFER and READ BUFFER commands by another application client. Buffer testing applications should insure that only a single application client is active. Use of reservations (to all logical units on the device) or linked commands may be helpful in avoiding buffer alteration between these two commands.

7.16 RECEIVE DIAGNOSTIC RESULTS command

The RECEIVE DIAGNOSTIC RESULTS command (see table 54) requests that data be sent to the application client after completion of a SEND DIAGNOSTIC command (see 7.23). If optional page formats are supported and the PCV bit is one, the page code field specifies the format of the returned data, and there is no relationship to a previous SEND DIAGNOSTIC command.

Table 54 – RECEIVE DIAGNOSTIC RESULTS command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (1Ch)							
1	Reserved							PCV
2	Page code							
3	(MSB) Allocation length							
4								(LSB)
5	Control							

If reservations are active, they shall affect the execution of the RECEIVE DIAGNOSTIC RESULTS command as follows. A reservation conflict shall occur when a RECEIVE DIAGNOSTIC RESULTS command is received from an initiator other than the one holding a logical unit reservation. If an initiator has an extent or element reservation on an SCSI device, and another initiator sends a RECEIVE DIAGNOSTIC RESULTS, a reservation conflict shall occur if the RECEIVE DIAGNOSTIC RESULTS affects the manner in which access to an extent or element reserved by the first initiator is performed. If the RECEIVE DIAGNOSTIC RESULTS does not affect access to the reserved extent or element, then a reservation conflict shall not occur.

A Page Code Valid (PCV) bit of zero indicates that the most recent SEND DIAGNOSTIC command shall define the data returned by this command. Optionally, a PCV bit of one indicates that the contents of the Page code field shall define the data returned by this command. Page code values are defined in 8.1 or in another command set standard (see 3.1.11).

NOTES

31 To insure that the diagnostic command information is not destroyed by a command sent from another initiator the logical unit should be reserved.

32 Although diagnostic software is generally device-specific, this command and the SEND DIAGNOSTIC command provide a means to isolate the operating system software from the device-specific diagnostic software. The operating system may remain device-independent.

See 8.1 for RECEIVE DIAGNOSTIC RESULTS page format definitions.

7.17 RELEASE(10) command

The RELEASE(10) command (see table 55) 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 55 — RELEASE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (57h)							
1	Reserved			3rdPty	Reserved		LongID	Extent
2	Reservation identification							
3	Third party device ID							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Parameter list length							
8	(LSB)							
9	Control							

The RESERVE and RELEASE commands provide a basic mechanism for contention resolution in multiple-initiator systems. See 5.3 for a general description of reservations and the commands that manage them. A reservation may only be released by a RELEASE command from the initiator 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 initiator. In this case, the device server shall return GOOD status without altering any other reservation.

If a device server has any reservation keys registered (see 7.13.1.1) a RELEASE command shall be rejected with a RESERVATION CONFLICT status. Reservation conflicts shall not occur for the RELEASE(10) command, except when reservation keys are registered.

7.17.1 Logical unit release (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 initiator to the specified logical unit. The reservation ID field in the command descriptor block shall be ignored by the device server.

7.17.2 Extent release (Optional)

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

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 7.21.2) is implemented.

7.17.3 Third-party release (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 7.21.3). Third-party release shall be implemented. It is intended for use in multiple-initiator 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 initiator ID, 3rdPty bit, and Third party device ID are identical when compared to the RESERVE command that established the reservation.

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

NOTE 33 If a target implements independent storage of mode parameters for each initiator, a third-party RESERVE command copies the current mode parameters for the initiator that sent the RESERVE command to the current mode parameters for the initiator specified as the third-party device (usually a copy master device). 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 change the third-party devices' current mode parameters back to their previous values. The third-party device may issue MODE SENSE and MODE SELECT commands to query and modify the mode parameters.

If the Third party device ID value associated with the reservation release is smaller than 255, the LongID bit may be zero and the ID value sent in the CDB. Device ID formats are protocol-specific. If the LongID bit is zero, the Parameter list length field shall be set to zero. If the Third party device ID is greater than 255, the LongID bit shall be one.

Device servers that support device IDs greater than 255 shall accept commands with LongID equal to one. Device servers whose devices IDs are limited to 255 or smaller may reject commands with LongID equal to one with CHECK CONDITION status and a sense key of ILLEGAL REQUEST.

If the LongID bit is one, the Parameter list length shall be eight, and the parameter list shall have the format shown in table 56. If the LongID bit is one, the Third party device ID field in the CDB shall be ignored. If the LongID bit is one and the Parameter list length is not eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

Table 56 — RELEASE(10) parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	Third party device ID							
7								(LSB)

7.18 RELEASE(6) command

The RELEASE(6) command (see table 57) 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. This clause describes only those instances where the RELEASE(6) command differs from the RELEASE(10) command. Except for the instances described in this clause, the RELEASE(6) command shall function exactly like the RELEASE(10) command (see 7.17).

Table 57 — RELEASE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (17h)							
1	Reserved			Obsolete				Extent
2	Reservation identification							
3	Reserved							
4	Reserved							
5	Control							

The RELEASE(6) command shall not release third-party reservations.

7.19 REPORT LUNS command

The REPORT LUNS command (see table 58) requests that the peripheral device logical unit numbers of known logical units in the target be sent to the application client. The REPORT LUNS command shall return information about only those logical units to which commands may be sent.

Table 58 — REPORT LUNS command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (A0h)							
1 – 5	Reserved							
6 – 9	(MSB) Allocation length (LSB)							
10	Reserved							
11	Control							

The REPORT LUNS command shall not be affected by reservations or persistent reservations.

The Allocation length shall be at least 16 bytes. If the Allocation length is less than 16 bytes, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to INVALID FIELD IN CDB.

The Allocation length is not sufficient to contain the logical unit number values for all configured logical units, the device server shall report as many logical unit number values as will fit in the specified Allocation length. This shall not be considered an error.

The device server shall report the logical unit numbers of configured logical units using the format shown in table 59.

Table 59 — LUN reporting parameter list format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3				LUN list length (n-7)				(LSB)
4								
7				Reserved				
				LUN list				
8	(MSB)							
15				LUN				(LSB)
				. . .				
n-7	(MSB)							
n				LUN				(LSB)

The LUN list length shall contain the length in bytes of the LUN list that is available to be transferred. The LUN list length is the number of logical unit numbers reported multiplied by eight. If the allocation length in the command descriptor block is too small to transfer information about all configured logical units, the LUN list length value shall not be adjusted to reflect the truncation.

7.20 REQUEST SENSE command

The REQUEST SENSE command (see table 60) requests that the device server transfer sense data to the application client.

Table 60 – REQUEST SENSE command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (03h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Allocation length							
5	Control							

The REQUEST SENSE command shall not be affected by reservations or persistent reservations.

Sense data shall be available and cleared under the conditions defined in SAM. If the device server has no other sense data available to return, it shall return a sense key of NO SENSE and an additional sense code of NO ADDITIONAL SENSE INFORMATION.

If the device server is in the Standby power condition or Idle power condition when a REQUEST SENSE command is received and there is no ACA condition, the device server shall return a sense key of NO SENSE and an additional sense code of LOW POWER CONDITION ON. On completion of the command the logical unit shall return to same power condition that was active before the REQUEST SENSE command was received. A REQUEST SENSE command shall not reset any active power condition timers.

The device server shall return CHECK CONDITION status for a REQUEST SENSE command only to report exception conditions specific to the command itself. For example:

a) An invalid filed value is detected in the command descriptor block;
b) An unrecovered parity error is detected by the service delivery subsystem;
c) A target malfunction that prevents return of the sense data.

If a recovered error occurs during the execution of the REQUEST SENSE command, the device server shall return the sense data with GOOD status. If a device server returns CHECK CONDITION status for a REQUEST SENSE command, the sense data may be invalid.

NOTE 34 The sense data appropriate to the selection of an invalid logical unit is defined in SAM.

Device servers shall be capable of returning eighteen bytes of data in response to a REQUEST SENSE command. If the allocation length is eighteen or greater, and a device server returns less than eighteen bytes of data, the application client should assume that the bytes not transferred would have been zeros had the device server returned those bytes. Application clients may determine how much sense data has been returned by examining the allocation length parameter in the command descriptor block and the additional sense length in the sense data. Device servers shall not adjust the additional sense length to reflect truncation if the allocation length is less than the sense data available.

The sense data format for response codes 70h (current errors) and 71h (deferred errors) are defined in table 61.

Table 61 − Response codes 70h and 71h sense data format

Bit / Byte	7	6	5	4	3	2	1	0
0	Valid	Response code (70h or 71h)						
1	Segment number							
2	Filemark	EOM	ILI	Reserved	Sense key			
3	(MSB)	Information						
6								(LSB)
7	Additional sense length (n-7)							
8	(MSB)	Command-specific information						
11								(LSB)
12	Additional sense code							
13	Additional sense code qualifier							
14	Field replaceable unit code							
15	SKSV	Sense-key specific						
17								
18	Additional sense bytes							
n								

A valid bit of zero indicates that the information field is not as defined in this standard. A valid bit of one indicates the information field contains valid information as defined in this standard. Device servers shall implement the valid bit.

Response code values 70h (current errors) is described in 7.20.2. Device servers shall implement Response code 70h. Response code value 71h (deferred errors) is described in 7.20.3. Implementation of Response code 71h is optional. Response code 7Fh is for a vendor-specific sense data formats. Response code values of 72h to 7Eh and 00h to 6Fh are reserved.

The segment number field contains the number of the current segment descriptor if the REQUEST SENSE command is in response to a COPY, COMPARE, or COPY AND VERIFY command. Up to 256 segments are supported, beginning with segment zero.

The Filemark bit is mandatory for sequential-access devices, and this bit is reserved for all other device types. A Filemark bit of one indicates that the current command has read a filemark or setmark. The additional sense code field may be used to indicate whether a filemark or setmark was read. Reporting of setmarks is optional and indicated by the Rsmk bit for sequential-access devices in the configuration parameters page. (See SSC.)

The end-of-medium (EOM) bit is mandatory for sequential-access and printer devices, and this bit is reserved for all other device types. An EOM bit of one indicates that an end-of-medium condition (end-of-partition, beginning-of-partition, out-of-paper, etc.) exists. For sequential-access devices, this bit indicates that the unit is at or past the early-warning if the direction was forward, or that the command was not completed because beginning-of-partition was encountered if the direction was reverse. (See SSC.)

An incorrect length indicator (ILI) bit of one usually indicates that the requested logical block length did not match the logical block length of the data on the medium.

The sense key, additional sense code and additional sense code qualifier provide a hierarchy of information. The intention of the hierarchy is to provide a top-down approach for an application client to determine information relating to the error and exception conditions. The sense key provides generic categories in which error and exception conditions may be reported. Application clients typically use sense keys for high level error recovery procedures. Additional sense codes provide further detail describing the sense key. Additional sense code qualifiers add further detail to the additional sense code. The additional sense code and additional sense code qualifier may be used by application clients where sophisticated error recovery procedures require detailed information describing the error and exception conditions.

The sense key field is mandatory and indicates generic information describing an error or exception condition. The sense keys are defined in 7.20.3.

The contents of the information field is device-type or command specific and is defined within the appropriate standard for the device type or command of interest. Device servers shall implement the information field. Unless specified otherwise, this field contains:

a) the unsigned logical block address associated with the sense key, for direct-access devices (device type 0), write-once devices (device type 4), CD-ROM devices (device type 5), and optical memory devices (device type 7);
b) the difference (residue) of the requested length minus the actual length in either bytes or blocks, as determined by the command, for sequential-access devices (device type 1), printer devices (device type 2), processor devices (device type 3) and some direct access device commands, except as defined for d) below. (Negative values are indicated by two's complement notation.);
c) the difference (residue) of the requested number of blocks minus the actual number of blocks copied or compared for the current segment descriptor of a COPY, COMPARE, or COPY AND VERIFY command; or
d) For sequential-access devices operating in buffered modes 1h or 2h that detect an unrecoverable write error when unwritten data blocks, filemarks, or setmarks remain in the buffer, the value of the information field for all commands shall be:
    1) the total number of data blocks, filemarks, and setmarks in the buffer if the device is in fixed block mode (block length field of the MODE SENSE block descriptor is non-zero and the fixed bit of the WRITE command is one); or
    2) the number of bytes in the buffer, including filemarks and setmarks, if the device is in variable mode (the fixed bit of the WRITE command is zero).

    For additional information see SSC.

The additional sense length field indicates the number of additional sense bytes to follow. If the allocation length of the command descriptor block is too small to transfer all of the additional sense bytes, the additional sense length is not adjusted to reflect the truncation.

The command-specific information field contains information that depends on the command that was executed. Further meaning for this field is defined within the command description. The command-specific information field is mandatory if the device server supports any of the following commands: COPY, COMPARE, COPY AND VERIFY, and REASSIGN BLOCKS.

The additional sense code (ASC) field indicates further information related to the error or exception condition reported in the sense key field. Device servers shall support the additional sense code field. Support of the additional sense codes not explicitly required by this standard is optional. A list of additional sense codes is in 7.20.3. If the device server does not have further information related to the error or exception condition, the additional sense code is set to NO ADDITIONAL SENSE INFORMATION.

The additional sense code qualifier (ASCQ) indicates detailed information related to the additional sense code. The additional sense code qualifier is optional. If the error or exception condition is reportable by the device, the value returned shall be

as specified in 7.20.3. If the device server does not have detailed information related to the error or exception condition, the additional sense code qualifier is set to zero.

Non-zero values in the field replaceable unit code field are used to define a device-specific mechanism or unit that has failed. A value of zero in this field shall indicate that no specific mechanism or unit has been identified to have failed or that the data is not available. The field replaceable unit code field is optional. The format of this information is not specified by this standard. Additional information about the field replaceable unit may be available in the ASCII information page (see 8.4.2), if supported by the device server.

The sense-key specific bytes are described in 7.20.1, below.

The additional sense bytes field may contain command specific data, peripheral device specific data, or vendor-specific data that further defines the nature of the CHECK CONDITION status.

7.20.1 Sense-key specific

The sense-key specific field as defined by this standard when the value of the sense-key specific valid (SKSV) bit is one. The sense-key specific valid bit and sense-key specific field are optional. The definition of this field is determined by the value of the sense key field. This field is reserved for sense keys not described below. An SKSV value of zero indicates that this field is not as defined by this standard.

If the sense key field is set to ILLEGAL REQUEST and the SKSV bit is set to one, the sense-key specific field shall be as defined as shown in table 62. The field pointer field indicates which illegal parameters in the command descriptor block or the data parameters are in error.

Table 62 — Field pointer bytes

Bit / Byte	7	6	5	4	3	2	1	0
15	SKSV	C/D	Reserved	Reserved	BPV	Bit pointer		
16	(MSB)	Field pointer						
17								(LSB)

A command data (C/D) bit of one indicates that the illegal parameter is in the command descriptor block. A C/D bit of zero indicates that the illegal parameter is in the data parameters sent by the application client in the Data-Out Buffer.

A bit pointer valid (BPV) bit of zero indicates that the value in the bit pointer field is not valid. A BPV bit of one indicates that the bit pointer field specifies which bit of the byte designated by the field pointer field is in error. When a multiple-bit field is in error, the bit pointer field shall point to the most-significant (left-most) bit of the field.

The field pointer field indicates which byte of the command descriptor block or of the parameter data was in error. Bytes are numbered starting from zero, as shown in the tables describing the commands and parameters. When a multiple-byte field is in error, the pointer shall point to the most- significant (left-most) byte of the field.

NOTE 35 Bytes identified as being in error are not necessarily the place that has to be changed to correct the problem.

If the sense key is RECOVERED ERROR, HARDWARE ERROR or MEDIUM ERROR and if the SKSV bit is one, the sense-key specific field shall be as shown in table 63.

Table 63 – Actual retry count bytes

Bit / Byte	7	6	5	4	3	2	1	0
15	SKSV	Reserved						
16	(MSB)	Actual retry count						
17								(LSB)

The actual retry count field returns vendor-specific information on the actual number of retries of the recovery algorithm used in attempting to recover an error or exception condition.

NOTE 36 This field should computed in the same way as the retry count fields within the error recovery page of the MODE SELECT command.

If the sense key is NOT READY or NO SENSE and the SKSV bit is one, the sense-key specific field shall be as shown in table 64.

Table 64 – Progress indication bytes

Bit / Byte	7	6	5	4	3	2	1	0
15	SKSV	Reserved						
16	(MSB)	Progress indication						
17								(LSB)

The progress indication field is a percent complete indication in which the returned value is the numerator that has 65536 (10000h) as its denominator. The progress indication shall be based upon the total operation.

NOTE 37 It is intended that the progress indication be time related. However, since for example format time varies with the number of defects encountered, etc., it is reasonable for the device server to assign values to various steps within the process. The granularity of these steps should be small enough to provide reasonable assurances to the application client that progress is being made.

7.20.2 Current errors

Response code 70h (current error) indicates that the CHECK CONDITION or COMMAND TERMINATED status returned is the result of an error or exception condition on the task that returned the CHECK CONDITION or COMMAND TERMINATED status or a protocol-specific failure condition. This includes errors generated during execution of the command. It also includes errors not related to any command that are first observed during execution of a command (e.g., disk servo-mechanism failure, off-track errors, and power-up test errors).

7.20.3 Deferred errors

Response code 71h (deferred error) indicates that the CHECK CONDITION status returned is the result of an error or exception condition that occurred during execution of a previous command for which GOOD status has already been returned. Such commands are associated with use of the immediate bit and with some forms of caching. Device servers that implement these features shall implement deferred error reporting.

The deferred error indication may be sent at a time selected by the device server through use of the asynchronous event reporting mechanism (see SAM), if AER is supported by both the application client and device server.

If AER is not supported, the deferred error may be indicated by returning CHECK CONDITION status to an application client on the appropriate initiator as described below. The subsequent execution of a REQUEST SENSE command shall return the deferred error sense information.

If the task terminates with CHECK CONDITION status and the subsequent sense data returns a deferred error that task shall not have been executed. After the device server detects a deferred error condition, it shall return a deferred error according to the rules described below:

a) If no external system intervention is necessary to recover a deferred error, a deferred error indication shall not be posted unless required by the error handling parameters of a MODE SELECT command. The occurrence of the error may be logged if statistical or error logging is supported;
b) If it is possible to associate a deferred error with a causing initiator and with a particular function or a particular subset of data, and the error is either unrecovered or required to be reported by the mode parameters, a deferred error indication shall be returned to an application client on the causing initiator. If an application client on an initiator other than the causing initiator attempts access to the particular function or subset of data associated with the deferred error, a BUSY status shall be returned to that application client in response to the command attempting the access;
c) If a deferred error cannot be associated with a causing initiator or with a particular subset of data, the device server shall return a deferred error indication to an application client on each initiator. If multiple deferred errors have accumulated for some initiators, only the last error shall be returned;
d) If a deferred error cannot be associated with a particular logical unit, the device server shall return a deferred error indication to an application client associated with any logical unit on the appropriate initiator; or
e) If a task has never entered the enabled task state, and a deferred error occurs, the task shall be terminated with CHECK CONDITION status and deferred error information posted in the sense data. If a deferred error occurs after a task has entered the enabled task state and the task is affected by the error, the task shall be terminated by CHECK CONDITION status and the current error information shall be returned in the sense data. In this case, if the current error information does not adequately define the deferred error condition, a deferred error may be returned after the current error information has been recovered. If a deferred error occurs after a task has entered the enabled task state and the task completes successfully, the device server may choose to return the deferred error information after the completion of the current command in conjunction with a subsequent command that has not started execution.

NOTE 38 Deferred errors may indicate that an operation was unsuccessful long after the command performing the data transfer returned GOOD status. If data that cannot be replicated or recovered from other sources is being stored using buffered write operations, synchronization commands should be performed before the critical data is destroyed in the host. This is necessary to be sure that recovery actions may be taken if deferred errors do occur in the storing of the data. If AER is not implemented, the synchronizing process should provide the necessary commands to allow returning CHECK CONDITION status and subsequent returning of deferred error sense information after all buffered operations are guaranteed to be complete.

7.20.4 Sense key and sense code definitions

The sense keys are defined in table 65.

Table 65 — Sense key descriptions

Sense key	Description
0h	NO SENSE. Indicates that there is no specific sense key information to be reported. This may occur for a successful command or for a command that receives CHECK CONDITION or COMMAND TERMINATED status because one of the Filemark, EOM, or ILI bits is set to one.
1h	RECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the device server. Details may be determinable by examining the additional sense bytes and the information field. When multiple recovered errors occur during one command, the choice of which error to report (first, last, most severe, etc.) is vendor-specific.
2h	NOT READY. Indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.
3h	MEDIUM ERROR. Indicates that the command terminated with a non-recovered error condition that was probably caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the device server is unable to distinguish between a flaw in the medium and a specific hardware failure (sense key 4h).
4h	HARDWARE ERROR. Indicates that the device server detected a non-recoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.
5h	ILLEGAL REQUEST. Indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands (FORMAT UNIT, SEARCH DATA, etc.). If the device server detects an invalid parameter in the command descriptor block, then it shall terminate the command without altering the medium. If the device server detects an invalid parameter in the additional parameters supplied as data, then the device server may have already altered the medium.
6h	UNIT ATTENTION. Indicates that the removable medium may have been changed or the target has been reset. See SAM for more detailed information about the unit attention condition.
7h	DATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.
8h	BLANK CHECK. Indicates that a write-once device or a sequential-access device encountered blank medium or format-defined end-of-data indication while reading or a write-once device encountered a non-blank medium while writing.

(continued)

Table 65 — Sense key descriptions (concluded)

Sense key	Description
9h	VENDOR-SPECIFIC. This sense key is available for reporting vendor specific conditions.
Ah	COPY ABORTED. Indicates a COPY, COMPARE, or COPY AND VERIFY command was aborted due to an error condition on the source device, the destination device, or both. (See 7.3.2 for additional information about this sense key.)
Bh	ABORTED COMMAND. Indicates that the device server aborted the command. The application client may be able to recover by trying the command again.
Ch	Obsolete
Dh	VOLUME OVERFLOW. Indicates that a buffered SCSI device has reached the end-of-partition and data may remain in the buffer that has not been written to the medium. One or more RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer. (See SSC.)
Eh	MISCOMPARE. Indicates that the source data did not match the data read from the medium.
Fh	RESERVED.

The additional sense codes and additional sense code qualifiers are defined in table 66.

Table 66 — ASC and ASCQ assignments

```
          D - DIRECT ACCESS DEVICE (SBC)                    Device Column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)               blank = code not
          . L - PRINTER DEVICE (SSC)                              used
          .  P - PROCESSOR DEVICE (SPC)                     not blank = code
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)               used
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC  ASCQ DTLPWRSOMCAE  DESCRIPTION
---  ---- ------------ ------------------------------------------------
67h  02h           A   ADD LOGICAL UNIT FAILED
13h  00h D   W  O      ADDRESS MARK NOT FOUND FOR DATA FIELD
12h  00h D   W  O      ADDRESS MARK NOT FOUND FOR ID FIELD
27h  03h  T            ASSOCIATED WRITE PROTECT
67h  06h           A   ATTACHMENT OF LOGICAL UNIT FAILED
00h  11h      R        AUDIO PLAY OPERATION IN PROGRESS
00h  12h      R        AUDIO PLAY OPERATION PAUSED
00h  14h      R        AUDIO PLAY OPERATION STOPPED DUE TO ERROR
00h  13h      R        AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED
```

(continued)

Table 66 – ASC and ASCQ assignments (continued)

ASC	ASCQ	DTLPWRSOMCAE	DESCRIPTION
66h	00h	S	AUTOMATIC DOCUMENT FEEDER COVER UP
66h	01h	S	AUTOMATIC DOCUMENT FEEDER LIFT UP
00h	04h	T S	BEGINNING-OF-PARTITION/MEDIUM DETECTED
0Ch	06h	DT W O	BLOCK NOT COMPRESSIBLE
14h	04h	T	BLOCK SEQUENCE ERROR
29h	03h	DTLPWRSOMCAE	BUS DEVICE RESET FUNCTION OCCURRED
11h	0Eh	DT WR O	CANNOT DECOMPRESS USING DECLARED ALGORITHM
30h	06h	DT W O	CANNOT FORMAT MEDIUM - INCOMPATIBLE MEDIUM
30h	02h	DT WR O	CANNOT READ MEDIUM - INCOMPATIBLE FORMAT
30h	01h	DT WR O	CANNOT READ MEDIUM - UNKNOWN FORMAT
30h	08h	R	CANNOT WRITE - APPLICATION CODE MISMATCH
30h	05h	DT WR O	CANNOT WRITE MEDIUM - INCOMPATIBLE FORMAT
30h	04h	DT WR O	CANNOT WRITE MEDIUM - UNKNOWN FORMAT
52h	00h	T	CARTRIDGE FAULT
73h	00h	R	CD CONTROL ERROR
3Fh	02h	DTLPWRSOMC	CHANGED OPERATING DEFINITION
11h	06h	WR O	CIRC UNRECOVERED ERROR
30h	03h	DT	CLEANING CARTRIDGE INSTALLED
30h	07h	DTL WRSOM AE	CLEANING FAILURE
00h	17h	DTL WRSOM AE	CLEANING REQUESTED
4Ah	00h	DTLPWRSOMCAE	COMMAND PHASE ERROR
2Ch	00h	DTLPWRSOMCAE	COMMAND SEQUENCE ERROR
6Eh	00h	A	COMMAND TO LOGICAL UNIT FAILED
2Fh	00h	DTLPWRSOMCAE	COMMANDS CLEARED BY ANOTHER INITIATOR
0Ch	04h	DT W O	COMPRESSION CHECK MISCOMPARE ERROR
67h	00h	A	CONFIGURATION FAILURE
67h	01h	A	CONFIGURATION OF INCAPABLE LOGICAL UNITS FAILED
2Bh	00h	DTLPWRSO C	COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT
67h	07h	A	CREATION OF LOGICAL UNIT FAILED
2Ch	04h	R	CURRENT PROGRAM AREA IS EMPTY
2Ch	03h	R	CURRENT PROGRAM AREA IS NOT EMPTY
30h	09h	R	CURRENT SESSION NOT FIXATED FOR APPEND
0Ch	05h	DT W O	DATA EXPANSION OCCURRED DURING COMPRESSION
69h	00h	A	DATA LOSS ON LOGICAL UNIT
41h	00h	D	DATA PATH FAILURE (SHOULD USE 40 NN)
4Bh	00h	DTLPWRSOMCAE	DATA PHASE ERROR
11h	07h	W O	DATA RE-SYNCHRONIZATION ERROR
16h	03h	D W O	DATA SYNC ERROR - DATA AUTO-REALLOCATED
16h	01h	D W O	DATA SYNC ERROR - DATA REWRITTEN
16h	04h	D W O	DATA SYNC ERROR - RECOMMEND REASSIGNMENT
16h	02h	D W O	DATA SYNC ERROR - RECOMMEND REWRITE
16h	00h	D W O	DATA SYNCHRONIZATION MARK ERROR
11h	0Dh	DT WR O	DE-COMPRESSION CRC ERROR
71h	00h	T	DECOMPRESSION EXCEPTION LONG ALGORITHM ID
70h	NNh	T	DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN
19h	00h	D O	DEFECT LIST ERROR
19h	03h	D O	DEFECT LIST ERROR IN GROWN LIST
19h	02h	D O	DEFECT LIST ERROR IN PRIMARY LIST
19h	01h	D O	DEFECT LIST NOT AVAILABLE
1Ch	00h	D O	DEFECT LIST NOT FOUND
32h	01h	D W O	DEFECT LIST UPDATE FAILURE
29h	04h	DTLPWRSOMCAE	DEVICE INTERNAL RESET
40h	NNh	DTLPWRSOMCAE	DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)
66h	02h	S	DOCUMENT JAM IN AUTOMATIC DOCUMENT FEEDER
66h	03h	S	DOCUMENT MISS FEED AUTOMATIC IN DOCUMENT FEEDER
72h	04h	R	EMPTY OR PARTIALLY WRITTEN RESERVED TRACK

(continued)

Table 66 – ASC and ASCQ assignments (continued)

ASC	ASCQ	DTLPWRSOMCAE	DESCRIPTION
34h	00h	DTLPWRSOMCAE	ENCLOSURE FAILURE
35h	00h	DTLPWRSOMCAE	ENCLOSURE SERVICES FAILURE
35h	03h	DTLPWRSOMCAE	ENCLOSURE SERVICES TRANSFER FAILURE
35h	04h	DTLPWRSOMCAE	ENCLOSURE SERVICES TRANSFER REFUSED
35h	02h	DTLPWRSOMCAE	ENCLOSURE SERVICES UNAVAILABLE
3Bh	0Fh	R	END OF MEDIUM REACHED
63h	00h	R	END OF USER AREA ENCOUNTERED ON THIS TRACK
00h	05h	T S	END-OF-DATA DETECTED
14h	03h	T	END-OF-DATA NOT FOUND
00h	02h	T S	END-OF-PARTITION/MEDIUM DETECTED
51h	00h	T O	ERASE FAILURE
0Ah	00h	DTLPWRSOMCAE	ERROR LOG OVERFLOW
11h	10h	R	ERROR READING ISRC NUMBER
11h	0Fh	R	ERROR READING UPC/EAN NUMBER
11h	02h	DT W SO	ERROR TOO LONG TO CORRECT
03h	02h	T	EXCESSIVE WRITE ERRORS
67h	04h	A	EXCHANGE OF LOGICAL UNIT FAILED
3Bh	07h	L	FAILED TO SENSE BOTTOM-OF-FORM
3Bh	06h	L	FAILED TO SENSE TOP-OF-FORM
5Dh	00h	DTLPWRSOMCAE	FAILURE PREDICTION THRESHOLD EXCEEDED
5Dh	FFh	DTLPWRSOMCAE	FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE)
00h	01h	T	FILEMARK DETECTED
14h	02h	T	FILEMARK OR SETMARK NOT FOUND
09h	02h	WR O	FOCUS SERVO FAILURE
31h	01h	D L R O	FORMAT COMMAND FAILED
58h	00h	O	GENERATION DOES NOT EXIST
1Ch	02h	D O	GROWN DEFECT LIST NOT FOUND
27h	01h	DT W O	HARDWARE WRITE PROTECTED
09h	04h	DT WR O	HEAD SELECT FAULT
00h	06h	DTLPWRSOMCAE	I/O PROCESS TERMINATED
10h	00h	D W O	ID CRC OR ECC ERROR
5Eh	03h	DTLPWRSO CA	IDLE CONDITION ACTIVATED BY COMMAND
5Eh	01h	DTLPWRSO CA	IDLE CONDITION ACTIVATED BY TIMER
22h	00h	D	ILLEGAL FUNCTION (USE 20 00, 24 00, OR 26 00)
64h	00h	R	ILLEGAL MODE FOR THIS TRACK
28h	01h	DT WR OM	IMPORT OR EXPORT ELEMENT ACCESSED
30h	00h	DT WR OM	INCOMPATIBLE MEDIUM INSTALLED
11h	08h	T	INCOMPLETE BLOCK READ
6Ah	00h	A	INFORMATIONAL, REFER TO LOG
48h	00h	DTLPWRSOMCAE	INITIATOR DETECTED ERROR MESSAGE RECEIVED
3Fh	03h	DTLPWRSOMCAE	INQUIRY DATA HAS CHANGED
44h	00h	DTLPWRSOMCAE	INTERNAL TARGET FAILURE
3Dh	00h	DTLPWRSOMCAE	INVALID BITS IN IDENTIFY MESSAGE
2Ch	02h	S	INVALID COMBINATION OF WINDOWS SPECIFIED
20h	00h	DTLPWRSOMCAE	INVALID COMMAND OPERATION CODE
21h	01h	DT WR OM	INVALID ELEMENT ADDRESS
24h	00h	DTLPWRSOMCAE	INVALID FIELD IN CDB
26h	00h	DTLPWRSOMCAE	INVALID FIELD IN PARAMETER LIST
49h	00h	DTLPWRSOMCAE	INVALID MESSAGE ERROR
64h	01h	R	INVALID PACKET SIZE
26h	04h	DTLPWRSOMCAE	INVALID RELEASE OF ACTIVE PERSISTENT RESERVATION
11h	05h	WR O	L-EC UNCORRECTABLE ERROR
60h	00h	S	LAMP FAILURE
5Bh	02h	DTLPWRSOM	LOG COUNTER AT MAXIMUM
5Bh	00h	DTLPWRSOM	LOG EXCEPTION
5Bh	03h	DTLPWRSOM	LOG LIST CODES EXHAUSTED
2Ah	02h	DTL WRSOMCAE	LOG PARAMETERS CHANGED

(continued)

Table 66 – ASC and ASCQ assignments (continued)

ASC	ASCQ	DTLPWRSOMCAE	DESCRIPTION
21h	00h	DT WR OM	LOGICAL BLOCK ADDRESS OUT OF RANGE
08h	03h	DT R OM	LOGICAL UNIT COMMUNICATION CRC ERROR (ULTRA-DMA/32)
08h	00h	DTL WRSOMCAE	LOGICAL UNIT COMMUNICATION FAILURE
08h	02h	DTL WRSOMCAE	LOGICAL UNIT COMMUNICATION PARITY ERROR
08h	01h	DTL WRSOMCAE	LOGICAL UNIT COMMUNICATION TIME-OUT
05h	00h	DTL WRSOMCAE	LOGICAL UNIT DOES NOT RESPOND TO SELECTION
4Ch	00h	DTLPWRSOMCAE	LOGICAL UNIT FAILED SELF-CONFIGURATION
3Eh	01h	A	LOGICAL UNIT FAILURE
3Eh	00h	DTLPWRSOMCAE	LOGICAL UNIT HAS NOT SELF-CONFIGURED YET
04h	01h	DTLPWRSOMCAE	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
68h	00h	A	LOGICAL UNIT NOT CONFIGURED
04h	00h	DTLPWRSOMCAE	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
04h	04h	DTL O	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS
04h	02h	DTLPWRSOMCAE	LOGICAL UNIT NOT READY, INITIALIZING CMD. REQUIRED
04h	08h	R	LOGICAL UNIT NOT READY, LONG WRITE IN PROGRESS
04h	03h	DTLPWRSOMCAE	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
04h	07h	DTLPWRSOMCAE	LOGICAL UNIT NOT READY, OPERATION IN PROGRESS
04h	05h	A	LOGICAL UNIT NOT READY, REBUILD IN PROGRESS
04h	06h	A	LOGICAL UNIT NOT READY, RECALCULATION IN PROGRESS
25h	00h	DTLPWRSOMCAE	LOGICAL UNIT NOT SUPPORTED
27h	02h	DT W O	LOGICAL UNIT SOFTWARE WRITE PROTECTED
5Eh	00h	DTLPWRSO CA	LOW POWER CONDITION ON
15h	01h	DTL WRSOM	MECHANICAL POSITIONING ERROR
53h	00h	DTL WRSOM	MEDIA LOAD OR EJECT FAILED
3Bh	0Dh	DT WR OM	MEDIUM DESTINATION ELEMENT FULL
31h	00h	DT WR O	MEDIUM FORMAT CORRUPTED
3Bh	13h	DT WR OM	MEDIUM MAGAZINE INSERTED
3Bh	14h	DT WR OM	MEDIUM MAGAZINE LOCKED
3Bh	11h	DT WR OM	MEDIUM MAGAZINE NOT ACCESSIBLE
3Bh	12h	DT WR OM	MEDIUM MAGAZINE REMOVED
3Bh	15h	DT WR OM	MEDIUM MAGAZINE UNLOCKED
3Ah	00h	DTL WRSOM	MEDIUM NOT PRESENT
3Ah	01h	DT WR OM	MEDIUM NOT PRESENT - TRAY CLOSED
3Ah	02h	DT WR OM	MEDIUM NOT PRESENT - TRAY OPEN
53h	02h	DT WR OM	MEDIUM REMOVAL PREVENTED
3Bh	0Eh	DT WR OM	MEDIUM SOURCE ELEMENT EMPTY
43h	00h	DTLPWRSOMCAE	MESSAGE ERROR
3Fh	01h	DTLPWRSOMCAE	MICROCODE HAS BEEN CHANGED
1Dh	00h	D W O	MISCOMPARE DURING VERIFY OPERATION
11h	0Ah	DT O	MISCORRECTED ERROR
2Ah	01h	DTL WRSOMCAE	MODE PARAMETERS CHANGED
67h	03h	A	MODIFICATION OF LOGICAL UNIT FAILED
69h	01h	A	MULTIPLE LOGICAL UNIT FAILURES
07h	00h	DTL WRSOM	MULTIPLE PERIPHERAL DEVICES SELECTED
11h	03h	DT W SO	MULTIPLE READ ERRORS
00h	00h	DTLPWRSOMCAE	NO ADDITIONAL SENSE INFORMATION
00h	15h	R	NO CURRENT AUDIO STATUS TO RETURN
32h	00h	D W O	NO DEFECT SPARE LOCATION AVAILABLE
11h	09h	T	NO GAP FOUND
01h	00h	D W O	NO INDEX/SECTOR SIGNAL
06h	00h	D WR OM	NO REFERENCE POSITION FOUND
02h	00h	D WR OM	NO SEEK COMPLETE
03h	01h	T	NO WRITE CURRENT
28h	00h	DTLPWRSOMCAE	NOT READY TO READY CHANGE, MEDIUM MAY HAVE CHANGED
00h	16h	DTLPWRSOMCAE	OPERATION IN PROGRESS
5Ah	01h	DT WR OM	OPERATOR MEDIUM REMOVAL REQUEST

(continued)

Table 66 – ASC and ASCQ assignments (continued)

ASC	ASCQ	DTLPWRSOMCAE	DESCRIPTION
5Ah	00h	DTLPWRSOM	OPERATOR REQUEST OR STATE CHANGE INPUT
5Ah	03h	DT W O	OPERATOR SELECTED WRITE PERMIT
5Ah	02h	DT W O	OPERATOR SELECTED WRITE PROTECT
61h	02h	S	OUT OF FOCUS
4Eh	00h	DTLPWRSOMCAE	OVERLAPPED COMMANDS ATTEMPTED
2Dh	00h	T	OVERWRITE ERROR ON UPDATE IN PLACE
63h	01h	R	PACKET DOES NOT FIT IN AVAILABLE SPACE
3Bh	05h	L	PAPER JAM
1Ah	00h	DTLPWRSOMCAE	PARAMETER LIST LENGTH ERROR
26h	01h	DTLPWRSOMCAE	PARAMETER NOT SUPPORTED
26h	02h	DTLPWRSOMCAE	PARAMETER VALUE INVALID
2Ah	00h	DTL WRSOMCAE	PARAMETERS CHANGED
69h	02h	A	PARITY/DATA MISMATCH
1Fh	00h	D O	PARTIAL DEFECT LIST TRANSFER
03h	00h	DTL W SO	PERIPHERAL DEVICE WRITE FAULT
27h	05h	T	PERMANENT WRITE PROTECT
27h	04h	T	PERSISTENT WRITE PROTECT
50h	02h	T	POSITION ERROR RELATED TO TIMING
3Bh	0Ch	T S	POSITION PAST BEGINNING OF MEDIUM
3Bh	0Bh	S	POSITION PAST END OF MEDIUM
15h	02h	DT WR O	POSITIONING ERROR DETECTED BY READ OF MEDIUM
73h	01h	R	POWER CALIBRATION AREA ALMOST FULL
73h	03h	R	POWER CALIBRATION AREA ERROR
73h	02h	R	POWER CALIBRATION AREA IS FULL
29h	01h	DTLPWRSOMCAE	POWER ON OCCURRED
29h	00h	DTLPWRSOMCAE	POWER ON, RESET, OR BUS DEVICE RESET OCCURRED
42h	00h	D	POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40 NN)
1Ch	01h	D O	PRIMARY DEFECT LIST NOT FOUND
73h	05h	R	PROGRAM MEMORY AREA IS FULL
73h	04h	R	PROGRAM MEMORY AREA UPDATE FAILURE
40h	00h	D	RAM FAILURE (SHOULD USE 40 NN)
15h	00h	DTL WRSOM	RANDOM POSITIONING ERROR
11h	11h	R	READ ERROR - LOSS OF STREAMING
3Bh	0Ah	S	READ PAST BEGINNING OF MEDIUM
3Bh	09h	S	READ PAST END OF MEDIUM
11h	01h	DT W SO	READ RETRIES EXHAUSTED
6Ch	00h	A	REBUILD FAILURE OCCURRED
6Dh	00h	A	RECALCULATE FAILURE OCCURRED
14h	01h	DT WR O	RECORD NOT FOUND
14h	06h	DT W O	RECORD NOT FOUND - DATA AUTO-REALLOCATED
14h	05h	DT W O	RECORD NOT FOUND - RECOMMEND REASSIGNMENT
14h	00h	DTL WRSO	RECORDED ENTITY NOT FOUND
18h	02h	D WR O	RECOVERED DATA - DATA AUTO-REALLOCATED
18h	05h	D WR O	RECOVERED DATA - RECOMMEND REASSIGNMENT
18h	06h	D WR O	RECOVERED DATA - RECOMMEND REWRITE
17h	05h	D WR O	RECOVERED DATA USING PREVIOUS SECTOR ID
18h	03h	R	RECOVERED DATA WITH CIRC
18h	07h	D W O	RECOVERED DATA WITH ECC - DATA REWRITTEN
18h	01h	D WR O	RECOVERED DATA WITH ERROR CORR. & RETRIES APPLIED
18h	00h	DT WR O	RECOVERED DATA WITH ERROR CORRECTION APPLIED
18h	04h	R	RECOVERED DATA WITH L-EC
17h	03h	DT WR O	RECOVERED DATA WITH NEGATIVE HEAD OFFSET
17h	00h	DT WRSO	RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
17h	02h	DT WR O	RECOVERED DATA WITH POSITIVE HEAD OFFSET
17h	01h	DT WRSO	RECOVERED DATA WITH RETRIES
17h	04h	WR O	RECOVERED DATA WITH RETRIES AND/OR CIRC APPLIED
17h	06h	D W O	RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED

(continued)

Table 66 – ASC and ASCQ assignments (continued)

ASC	ASCQ	DTLPWRSOMCAE	DESCRIPTION
17h	09h	D W O	RECOVERED DATA WITHOUT ECC - DATA REWRITTEN
17h	07h	D W O	RECOVERED DATA WITHOUT ECC - RECOMMEND REASSIGNMENT
17h	08h	D W O	RECOVERED DATA WITHOUT ECC - RECOMMEND REWRITE
1Eh	00h	D W O	RECOVERED ID WITH ECC CORRECTION
6Bh	01h	A	REDUNDANCY LEVEL GOT BETTER
6Bh	02h	A	REDUNDANCY LEVEL GOT WORSE
67h	05h	A	REMOVE OF LOGICAL UNIT FAILED
3Bh	08h	T	REPOSITION ERROR
2Ah	03h	DTLPWRSOMCAE	RESERVATIONS PREEMPTED
36h	00h	L	RIBBON, INK, OR TONER FAILURE
37h	00h	DTL WRSOMCAE	ROUNDED PARAMETER
5Ch	00h	D O	RPL STATUS CHANGE
39h	00h	DTL WRSOMCAE	SAVING PARAMETERS NOT SUPPORTED
62h	00h	S	SCAN HEAD POSITIONING ERROR
29h	02h	DTLPWRSOMCAE	SCSI BUS RESET OCCURRED
47h	00h	DTLPWRSOMCAE	SCSI PARITY ERROR
54h	00h	P	SCSI TO HOST SYSTEM INTERFACE FAILURE
45h	00h	DTLPWRSOMCAE	SELECT OR RESELECT FAILURE
3Bh	00h	TL	SEQUENTIAL POSITIONING ERROR
72h	00h	R	SESSION FIXATION ERROR
72h	03h	R	SESSION FIXATION ERROR - INCOMPLETE TRACK IN SESSION
72h	01h	R	SESSION FIXATION ERROR WRITING LEAD-IN
72h	02h	R	SESSION FIXATION ERROR WRITING LEAD-OUT
00h	03h	T	SETMARK DETECTED
3Bh	04h	L	SLEW FAILURE
09h	03h	WR O	SPINDLE SERVO FAILURE
5Ch	02h	D O	SPINDLES NOT SYNCHRONIZED
5Ch	01h	D O	SPINDLES SYNCHRONIZED
5Eh	04h	DTLPWRSO CA	STANDBY CONDITION ACTIVATED BY COMMAND
5Eh	02h	DTLPWRSO CA	STANDBY CONDITION ACTIVATED BY TIMER
6Bh	00h	A	STATE CHANGE HAS OCCURRED
1Bh	00h	DTLPWRSOMCAE	SYNCHRONOUS DATA TRANSFER ERROR
55h	01h	D O	SYSTEM BUFFER FULL
55h	00h	P	SYSTEM RESOURCE FAILURE
4Dh	NNh	DTLPWRSOMCAE	TAGGED OVERLAPPED COMMANDS (NN = QUEUE TAG)
33h	00h	T	TAPE LENGTH ERROR
3Bh	03h	L	TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY
3Bh	01h	T	TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM
3Bh	02h	T	TAPE POSITION ERROR AT END-OF-MEDIUM
3Fh	00h	DTLPWRSOMCAE	TARGET OPERATING CONDITIONS HAVE CHANGED
5Bh	01h	DTLPWRSOM	THRESHOLD CONDITION MET
26h	03h	DTLPWRSOMCAE	THRESHOLD PARAMETERS NOT SUPPORTED
3Eh	02h	A	TIMEOUT ON LOGICAL UNIT
2Ch	01h	S	TOO MANY WINDOWS SPECIFIED
09h	00h	DT WR O	TRACK FOLLOWING ERROR
09h	01h	WR O	TRACKING SERVO FAILURE
61h	01h	S	UNABLE TO ACQUIRE VIDEO
57h	00h	R	UNABLE TO RECOVER TABLE-OF-CONTENTS
53h	01h	T	UNLOAD TAPE FAILURE
11h	00h	DT WRSO	UNRECOVERED READ ERROR
11h	04h	D W O	UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED
11h	0Bh	D W O	UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT
11h	0Ch	D W O	UNRECOVERED READ ERROR - RECOMMEND REWRITE THE DATA
46h	00h	DTLPWRSOMC	UNSUCCESSFUL SOFT RESET
35h	01h	DTLPWRSOMCAE	UNSUPPORTED ENCLOSURE FUNCTION
59h	00h	O	UPDATED BLOCK READ

(continued)

Table 66 − ASC and ASCQ assignments (concluded)

ASC	ASCQ	DTLPWRSOMCAE	DESCRIPTION
61h	00h	S	VIDEO ACQUISITION ERROR
65h	00h	DTLPWRSOMCAE	VOLTAGE FAULT
0Bh	00h	DTLPWRSOMCAE	WARNING
0Bh	02h	DTLPWRSOMCAE	WARNING - ENCLOSURE DEGRADED
0Bh	01h	DTLPWRSOMCAE	WARNING - SPECIFIED TEMPERATURE EXCEEDED
50h	00h	T	WRITE APPEND ERROR
50h	01h	T	WRITE APPEND POSITION ERROR
0Ch	00h	T RS	WRITE ERROR
0Ch	02h	D W O	WRITE ERROR - AUTO REALLOCATION FAILED
0Ch	09h	R	WRITE ERROR - LOSS OF STREAMING
0Ch	0Ah	R	WRITE ERROR - PADDING BLOCKS ADDED
0Ch	03h	D W O	WRITE ERROR - RECOMMEND REASSIGNMENT
0Ch	01h		WRITE ERROR - RECOVERED WITH AUTO REALLOCATION
0Ch	08h	R	WRITE ERROR - RECOVERY FAILED
0Ch	07h	R	WRITE ERROR - RECOVERY NEEDED
27h	00h	DT W O	WRITE PROTECTED
80h	*xx*h	\	
THROUGH		>	Vendor-specific.
FFh	*xx*h	/	
*xx*h	80h	\	
THROUGH		>	Vendor-specific QUALIFICATION OF STANDARD ASC.
*xx*h	FFh	/	
			ALL CODES NOT SHOWN ARE RESERVED.

NOTE - Annex B contains the ASC and ASCQ assignments in numeric order.

7.21 RESERVE(10) command

The RESERVE(10) command (see table 67) is used to reserve a logical unit or, if the extent reservation option is implemented, extents within a logical unit.

Table 67 — RESERVE(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (56h)							
1	Reserved			3rdPty	Reserved		LongID	Extent
2	Reservation identification							
3	Third party device ID							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB) Parameter list length							
8	(LSB)							
9	Control							

The RESERVE and RELEASE commands provide the basic mechanism for contention resolution in multiple-initiator systems. The third-party reservation allows logical units or extents to be reserved for another specified SCSI device. See 5.3 for a general description of reservations and the commands that manage them.

If the RESERVE(10) command is implemented, then the RELEASE(10) also shall be implemented.

If a device server has any reservation keys registered (see 7.13.1.1) a RESERVE command shall be rejected with a RESERVATION CONFLICT status.

7.21.1 Logical unit reservation (Mandatory)

If the extent bit is zero, this command shall request that the entire logical unit be reserved for the exclusive use of the initiator until the reservation is superseded by another valid RESERVE command from the same initiator or until released by a RELEASE command from the same initiator that made the reservation, by a TARGET RESET task management function performed by any initiator, 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 initiator. It shall be permissible for an initiator to reserve a logical unit that is currently reserved by that initiator. If the extent bit is zero, the reservation identification and the Parameter list length shall be ignored.

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

After honoring a logical unit reservation, the device server shall check each newly received command for reservation conflicts. See 5.3.1.

For multiple port implementations, devices on other ports (i.e., the ports that do not include the initiator to which the reservation has been granted) also shall be denied access to the logical unit as described in the preceding paragraph.

7.21.2 Extent reservation (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 7.18.2) shall also be implemented. These options permit multiple extents within the logical unit to be reserved, each with a separate reservation type. (Reservation types are listed in table 69.)

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 Parameter 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 initiator that made the reservation or until released by a RELEASE command from the same initiator, by a TARGET RESET task management function performed by any initiator, by a hard reset condition, or by a power on cycle. (7.21.4 describes how reservations may be superseded.) If any of the last three conditions occur, a unit attention condition shall be generated; and
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 Parameter list length field. The extent list shall consist of zero or more descriptors as shown in table 68. 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 68 – Data format of extent descriptors

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

The reservation type field shall define the type of reservation in effect for the extent. The types of reservation are defined in table 69.

Table 69 – 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 initiator shall be permitted read operations to the indicated extent. This reservation shall not inhibit write operations from any initiator 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 initiator shall be permitted write operations to the indicated extent. This reservation shall not inhibit read operations from any initiator or conflict with a read exclusive reservation from any initiator. 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 initiator 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 initiator to the indicated extent. This reservation shall not inhibit read operations from any initiator 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, a reservation conflict shall occur. (See 5.3.1.)

7.21.3 Third-party reservation (Mandatory)

The third-party reservation for the RESERVE(10) command allows an application client to reserve a logical unit or extents within a logical unit for another SCSI device. This is intended for use in multiple initiator systems that use the COPY command.

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 device specified in the third-party device ID field. Device ID formats are protocol-specific. The device server shall preserve the reservation until it is superseded by another valid RESERVE command from the initiator that made the reservation or until it is released by the same initiator, by a TARGET RESET task management function performed by any initiator, a hard reset condition, or by a power on cycle. The device server shall ignore any attempt to release the reservation made by any other initiator.

After a third-party reservation has been granted, the initiator that sent the RESERVE command shall be treated like any other initiator. Reservation conflicts shall occur in all cases where another initiator is not allowed access due to the reservation.

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

NOTE 39 This transfer of the mode parameters is applicable to device servers that store mode information independently for different initiators. This mechanism allows an application client to set the mode parameters of a target for the use of a copy master (i.e., the third-party device). The third-party copy master may subsequently issue a MODE SELECT command to modify the mode parameters.

If the Third party device ID value associated with the reservation release is smaller than 255, the LongID bit may be zero and the ID value sent in the CDB. Device ID formats are protocol-specific. If the Third party device ID is greater than 255, the LongID bit shall be one. If the LongID bit is one, the Third party device ID field in the CDB shall be ignored. If the LongID bit is one, the Parameter list length shall be at least eight. If the LongID bit is one and the Parameter list length is less than eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

Device servers that support device IDs greater than 255 shall accept commands with LongID equal to one. Device servers whose devices IDs are limited to 255 or smaller may reject commands with LongID equal to one with CHECK CONDITION status and a sense key of ILLEGAL REQUEST.

If both the LongID and Extent bits are one, then the parameter list shall have the format shown in table 70 and the extent list length shall be the Parameter list length minus eight.

Table 70 – RESERVE(10) ID & extents parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7	Third party device ID							(LSB)
8	Extent descriptors (see table 68)							
n								

If the LongID bit is one and the Extent bit is zero, the Parameter list length shall be eight, and the parameter list shall have the format shown in table 71. If the LongID bit is one and the Extent bit is zero and the Parameter list length is not eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

Table 71 − RESERVE(10) ID only parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	Third party device ID							
7								(LSB)

If the LongID bit is zero, the Parameter list shall be processed as an extent list. (See 7.21.2.)

7.21.4 Superseding reservations (Mandatory)

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 40 Superseding reservations allow the SCSI device ID in a third-party reservation to be changed. This capability is necessary for certain situations when using COMPARE, COPY, and COPY AND VERIFY commands.

7.22 RESERVE(6) command

The RESERVE(6) command (see table 72) is used to reserve a logical unit or, if the extent reservation option is implemented, extents within a logical unit. This clause describes only those instances where the RESERVE(6) command differs from the RESERVE(10) command. Except for the instances described in this clause, the RESERVE(6) command shall function exactly like the RESERVE(10) command (see 7.21).

Table 72 — RESERVE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (16h)							
1	Reserved				Obsolete			Extent
2	Reservation identification							
3	(MSB)							
4	Parameter list length							(LSB)
5	Control							

If the RESERVE(6) command is implemented, then the RELEASE(6) also shall be implemented.

The RESERVE(6) command shall not allow third party reservations.

7.23 SEND DIAGNOSTIC command

The SEND DIAGNOSTIC command (see table 73) requests the device server to perform diagnostic operations on the target, on the logical unit, or on both. The only mandatory implementation of this command is the self-test feature with the parameter list length of zero. Except when the SelfTest bit is one, this command is usually followed by a RECEIVE DIAGNOSTIC RESULTS (see 7.16) command.

Table 73 — SEND DIAGNOSTIC command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (1Dh)							
1	Reserved			PF	Reserved	SelfTest	DevOfL	UnitOfL
2	Reserved							
3	(MSB)							
4	Parameter list length							(LSB)
5	Control							

If reservations are active, they shall affect the execution of the SEND DIAGNOSTIC command as follows. A reservation conflict shall occur when a SEND DIAGNOSTIC command is received from an initiator other than the one holding a logical

unit reservation. If an initiator has an extent or element reservation on an SCSI device, and an another initiator sends a SEND DIAGNOSTIC, a reservation conflict shall occur if the SEND DIAGNOSTIC affects the manner in which access to an extent or element reserved by the first initiator is performed. If the SEND DIAGNOSTIC does not affect access to the reserved extent or element, then a reservation conflict shall not occur.

A page format (PF) bit of one specifies that the SEND DIAGNOSTIC parameters conform to the page structure as specified in this standard. The implementation of the PF bit is optional. See 8.1 for the definition of diagnostic pages. A PF bit of zero indicates that all SEND DIAGNOSTIC parameters vendor-specific.

A self-test (SelfTest) bit of one directs the device server to complete the target's default self-test. If the self-test successfully passes, the command shall be terminated with GOOD status; otherwise, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to HARDWARE ERROR.

A self-test bit of zero requests that the device server perform the diagnostic operation specified in the parameter list. The diagnostic operation might or might not require the device server to return data that contains diagnostic results. If the return of data is not required, the return of GOOD status indicates successful completion of the diagnostic operation. If the return of data is required, the device server shall either:

a) perform the requested diagnostic operation, prepare the data to be returned and indicate completion by returning GOOD status. The application client issues a RECEIVE DIAGNOSTIC RESULTS command to recover the data; or
b) accept the parameter list, and if no errors are detected in the parameter list, return GOOD status. The requested diagnostic operation and the preparation of the data to be returned are performed upon receipt of a RECEIVE DIAGNOSTIC RESULTS command.

A UnitOfL bit of one grants permission to the device server to perform diagnostic operations that may affect the user accessible medium on the logical unit, e.g., write operations to the user accessible medium, or repositioning of the medium on sequential access devices. The implementation of the UnitOfl bit is optional. A UnitOfL bit of zero prohibits any diagnostic operations that may be detected by subsequent tasks.

A DevOfl bit of one grants permission to the device server to perform diagnostic operations that may affect all the logical units on a target, e.g., alteration of reservations, log parameters, or sense data. The implementation of the DevOfl bit is optional. A DevOfL bit of zero prohibits diagnostic operations that may be detected by subsequent tasks.

The parameter list length field specifies the length in bytes of the parameter list that shall be transferred from the application client to the device server. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered an error. If the specified parameter list length results in the truncation of one or more pages (PF bit set to one) the device server shall return CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

NOTE 41 To insure that the diagnostic command information is not destroyed by a command sent from another initiator, either the SEND DIAGNOSTIC command should be linked to the RECEIVE DIAGNOSTIC RESULTS command or the logical unit should be reserved.

7.24 TEST UNIT READY Command

The TEST UNIT READY command (see table 74) provides a means to check if the logical unit is ready. This is not a request for a self-test. If the logical unit is able to accept an appropriate medium-access command without returning CHECK CONDITION status, this command shall return a GOOD status. If the logical unit cannot become operational or is in a state such that an application client action (e.g., START UNIT command) is required to make the unit ready, the device server shall return CHECK CONDITION status with a sense key of NOT READY.

Table 74 — TEST UNIT READY command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (00h)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Control							

If reservations are active, they shall affect the execution of the TEST UNIT READY command as follows. A reservation conflict shall occur when a TEST UNIT READY command is received from an initiator other than the one holding a logical unit reservation. The TEST UNIT READY command shall not be affected by extent or element reservations.

Table 75 defines the suggested GOOD and CHECK CONDITION status responses to the TEST UNIT READY command. Other conditions, including deferred errors, may result in other responses (e.g., BUSY or RESERVATION CONFLICT status).

Table 75 — Preferred TEST UNIT READY responses

Status	Sense key	ASC and ASCQ
GOOD	NO SENSE	NO ADDITIONAL SENSE INFORMATION or other valid additional sense code.
CHECK CONDITION	ILLEGAL REQUEST	LOGICAL UNIT NOT SUPPORTED
CHECK CONDITION	NOT READY	LOGICAL UNIT DOES NOT RESPOND TO SELECTION
CHECK CONDITION	NOT READY	MEDIUM NOT PRESENT
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
CHECK CONDITION	NOT READY	LOGICAL UNIT IS IN PROCESS OF BECOMING READY

(continued)

Table 75 − Preferred TEST UNIT READY responses (concluded)

Status	Sense key	ASC and ASCQ
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS

7.25 WRITE BUFFER command

The WRITE BUFFER command (see table 76) is used in conjunction with the READ BUFFER command as a diagnostic function for testing logical unit memory in the target SCSI device and the integrity of the service delivery subsystem. Additional modes are provided for downloading microcode and for downloading and saving microcode.

Table 76 − WRITE BUFFER command

Bit / Byte	7	6	5	4	3	2	1	0
0	Operation code (3Bh)							
1	Reserved					Mode		
2	Buffer ID							
3	(MSB)							
4	Buffer offset							
5								(LSB)
6	(MSB)							
7	Parameter list length							
8								(LSB)
9	Control							

If reservations are active, they shall affect the execution of the WRITE BUFFER command as follows. A reservation conflict shall occur when a WRITE BUFFER command is received from an initiator other than the one holding a logical unit, extent, or element reservation.

This command shall not alter any medium of the logical unit when the data mode or the combined header and data mode is specified.

The function of this command and the meaning of fields within the command descriptor block depend on the contents of the Mode field. The Mode field is defined in table 77.

Table 77 — WRITE BUFFER Mode field

Mode	Description	Implementation requirements
000b	Write combined header and data	Optional
001b	Vendor-specific	Vendor-specific
010b	Write data	Optional
011b	Reserved	Reserved
100b	Download microcode	Optional
101b	Download microcode and save	Optional
110b	Download microcode with offsets	Optional
111b	Download microcode with offsets and save	Optional

NOTES
42 Modes 000b and 001b are not recommended.
43 When downloading microcode with buffer offsets, the WRITE BUFFER command mode should be 110b or 111b.

7.25.1 Combined header and data mode (000b)

In this mode, data to be transferred is preceded by a four-byte header. The four-byte header consists of all reserved bytes. The buffer ID and the buffer offset fields shall be zero. The parameter list length field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer. This number includes four bytes of header, so the data length to be stored in the device server's buffer is parameter list length minus four. The application client should attempt to ensure that the parameter list length is not greater than four plus the buffer capacity (see 7.15.1) that is returned in the header of the READ BUFFER command (mode 00b). If the parameter list length exceeds the buffer capacity device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST.

7.25.2 Vendor-specific mode (001b)

In this mode, the meaning of the buffer ID, buffer offset, and parameter list length fields are not specified by this standard.

7.25.3 Data mode (010b)

In this mode, the Data-Out Buffer contains buffer data destined for the logical unit. The buffer ID field identifies a specific buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. Buffer ID zero shall be supported. If more than one buffer is supported, additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is selected, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

Data are written to the logical unit buffer starting at the location specified by the buffer offset. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor. If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status and it shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The parameter list length specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should attempt to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. (The capacity of the buffer may be determined by the buffer capacity field in the READ BUFFER descriptor.) If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

7.25.4 Download microcode mode (100b)

If the logical unit cannot accept this command because of some device condition, the device server shall terminate each WRITE BUFFER command with this mode (100b) with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

In this mode, vendor-specific microcode or control information shall be transferred to the control memory space of the logical unit. After a power-cycle or reset, the device operation shall revert to a vendor-specific condition. The meanings of the buffer ID, buffer offset, and parameter list length fields are not specified by this standard and are not required to be zero-filled. When the microcode download has completed successfully the device server shall generate a unit attention condition for all initiators except the one that issued the WRITE BUFFER command (see SAM). The additional sense code shall be set to MICROCODE HAS BEEN CHANGED.

7.25.5 Download microcode and save mode (101b)

If the logical unit cannot accept this command because of some device condition, the device server shall terminate each WRITE BUFFER command with this mode (101b) with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

In this mode, vendor-specific microcode or control information shall be transferred to the logical unit and, if the WRITE BUFFER command is completed successfully, also shall be saved in a non-volatile memory space (semiconductor, disk, or other). The downloaded code shall then be effective after each power-cycle and reset until it is supplanted in another download microcode and save operation. The meanings of the buffer ID, buffer offset, and parameter list length fields are not specified by this standard and are not required to be zero-filled. When the download microcode and save command has completed successfully the device server shall generate a unit attention condition (see SAM) for all initiators except the one that issued the WRITE BUFFER command. When reporting the unit attention condition, the device server shall set the additional sense code to MICROCODE HAS BEEN CHANGED.

7.25.6 Download microcode with offsets (110b)

In this mode, the application client may split the transfer of the vendor-specific microcode or control information over two or more WRITE BUFFER commands. If the logical unit cannot accept this command because of some device condition, the device server shall terminate each WRITE BUFFER command with this mode (110b) with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

If the last WRITE BUFFER command of a set of one or more commands completes successfully, the microcode or control information shall be transferred to the control memory space of the logical unit. After a power-cycle or reset, the device shall revert to a vendor-specific condition. In this mode, the Data-Out Buffer contains vendor-specific, self-describing microcode or control information.

Since the downloaded microcode or control information may be sent using several commands, when the logical unit detects the last download microcode with offsets and save mode WRITE BUFFER command has been received, the device server shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command. After the last command completes successfully the device server shall generate a unit attention condition (see SAM) for all initiators except the one that issued the set of WRITE BUFFER commands. When reporting the unit attention condition, the device server shall set the additional sense code to MICROCODE HAS BEEN CHANGED.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (one or more commands) are not received before a reset or power-on cycle occurs, the change shall not be effective and the new microcode or control information shall be discarded.

The buffer ID field identifies a specific buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. A Buffer ID value of zero shall be supported. If more than one buffer is supported, additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is identified, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The microcode or control information are written to the logical unit buffer starting at the location specified by the buffer offset. The application client shall send commands that conform to the offset boundary requirements (see 7.15.4). If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status and it shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The parameter list length specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should attempt to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. (The capacity of the buffer may be determined by the buffer capacity field in the READ BUFFER descriptor.) If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

7.25.7 Download microcode with offsets and save mode (111b)

In this mode, the initiator may split the transfer of the vendor-specific microcode or control information over two or more WRITE BUFFER commands. If the logical unit cannot accept this command because of some device condition, the device server shall terminate each mode 111b WRITE BUFFER command with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and shall set the additional sense code to COMMAND SEQUENCE ERROR.

If the last WRITE BUFFER command of a set of one or more commands completes successfully, the microcode or control information shall be saved in a non-volatile memory space (semiconductor, disk, or other). The saved downloaded microcode or control information shall then be effective after each power-cycle and reset until it is supplanted by another download microcode with save operation or download microcode with offsets and save operation. In this mode, the Data-Out Buffer contains vendor-specific, self-describing microcode or control information.

Since the downloaded microcode or control information may be sent using several commands, when the logical unit detects the last download microcode with offsets and save mode WRITE BUFFER command has been received, the device server shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command. After the last command completes successfully the device server shall generate a unit attention condition (see SAM) for all initiators except the one that issued the set of WRITE BUFFER commands. When reporting the unit attention condition, the device server shall set the additional sense code to MICROCODE HAS BEEN CHANGED.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (one or more commands) are not received before a reset or power-on cycle occurs, the change shall not be effective and the new microcode or control information shall be discarded.

The buffer ID field identifies a specific buffer within the logical unit. The vendor assigns buffer ID codes to buffers within the logical unit. A Buffer ID value of zero shall be supported. If more than one buffer is supported, additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is identified, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The microcode or control information are written to the logical unit buffer starting at the location specified by the buffer offset. The application client shall conform to the offset boundary requirements. If the device server is unable to accept the specified buffer offset, it shall return CHECK CONDITION status and it shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

The parameter list length specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should attempt to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. (The capacity of the buffer may be determined by the buffer capacity field in the READ BUFFER descriptor.) If the buffer offset and parameter list length fields specify a transfer in excess of the buffer capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

8 Parameters for all device types

8.1 Diagnostic parameters

This clause describes the diagnostic page structure and the diagnostic pages that are applicable to all SCSI devices. Pages specific to each device type are described in the command standard (see 3.1.11) that applies to that device type.

A SEND DIAGNOSTIC command with a PF bit of one specifies that the SEND DIAGNOSTIC parameter list consists of zero or more diagnostic pages and that the data returned by the subsequent RECEIVE DIAGNOSTIC RESULTS command shall use the diagnostic page format defined in table 78. A RECEIVE DIAGNOSTIC RESULTS command with a PCV bit of one specifies that the device server return a diagnostic page using the format defined in table 78.

Table 78 — Diagnostic page format

Bit / Byte	7	6	5	4	3	2	1	0
0	Page code							
1	Reserved							
2	(MSB)			Page length (n-3)				
3								(LSB)
4	Diagnostic parameters							
n								

Each diagnostic page defines a function or operation that the device server shall perform (SEND DIAGNOSTIC command) or the information being returned (RECEIVE DIAGNOSTIC RESULTS with PCV equal to one). The page contains a page header followed by the data that is formatted according to the page code specified.

Device servers that implement diagnostic pages are only required to accept a single diagnostic page per command.

The page code field identifies which diagnostic page is being sent (SEND DIAGNOSTIC), requested (RECEIVE DIAGNOSTIC RESULTS with PCV equal to one) or returned (RECEIVE DIAGNOSTIC RESULTS parameter data). The page codes are defined in table 79.

Table 79 − Diagnostic page codes

Page code	Description	Clause
00h	Supported diagnostics pages	8.1.1
01h - 3Fh	Reserved for pages that apply to all device types	
40h - 7Fh	See specific device type for definition	
80h - FFh	Vendor-specific pages	

The page length field specifies the length in bytes of the diagnostic parameters that follow this field. If the application client sends a page length that results in the truncation of any parameter, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The diagnostic parameters are defined for each page code. The diagnostic parameters within a page may be defined differently in a SEND DIAGNOSTIC command than in a RECEIVE DIAGNOSTIC RESULTS command.

8.1.1 Supported diagnostic pages

The supported diagnostics page (see table 80) returns the list of diagnostic pages implemented by the device server. This page shall be implemented if the device server implements the page format option of the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands.

Table 80 − Supported diagnostic pages

Bit / Byte	7	6	5	4	3	2	1	0
0	Page code (00h)							
1	Reserved							
2	(MSB)			Page length (n-3)				
3								(LSB)
4	Supported page list							
n								

The definition of this page for the SEND DIAGNOSTIC command includes only the first four bytes. If the page length field is not zero, the device server shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN PARAMETER LIST. This page instructs the device server to make available the list of all supported diagnostic pages to be returned by a subsequent RECEIVE DIAGNOSTIC RESULTS command.

The definition of this page for the RECEIVE DIAGNOSTIC RESULTS command includes the list of diagnostic pages supported by the device server.

The page length field specifies the length in bytes of the following supported page list.

The supported page list field shall contain a list of all diagnostic page codes implemented by the device server in ascending order beginning with page code 00h.

8.2 Log parameters

This clause describes the log page structure and the log pages that are applicable to all SCSI devices. Pages specific to each device type are described in the command standard (see 3.1.11) that applies to that device type. The LOG SELECT command supports the ability to send zero or more log pages. The LOG SENSE command (see 7.7) returns a single log page specified in the page code field of the command descriptor block.

Each log page begins with a four-byte page header followed by zero or more variable-length log parameters defined for that page. The log page format is defined in table 81.

Table 81 – Log page format

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved		Page code					
1	Reserved							
2	(MSB)							
3	Page length (n-3)							(LSB)
	Log parameters(s)							
4	Log parameter (First) (Length x)							
x+3								
	. .							
n-y+1	Log parameter (Last) (Length y)							
n								

The page code field identifies which log page is being transferred.

The page length field specifies the length in bytes of the following log parameters. If the application client sends a page length that results in the truncation of any parameter, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Most log pages contain one or more special data structures called log parameters (see table 82). Log parameters may be data counters of a particular event (or events), the conditions under which certain operations were performed, or list parameters (strings) that contain a description of a particular event.

Table 82 − Log parameter

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1	Parameter code							(LSB)
2	DU	DS	TSD	ETC	TMC		LBIN	LP
3	Parameter length (n-3)							
4	Parameter value							
n								

Each log parameter begins with a four-byte parameter header followed by one or more bytes of parameter value data.

The parameter code field identifies the log parameter being transferred for that log page.

The DU, DS, TSD, ETC, TMC, LBIN, and LP fields are collectively referred to as the parameter control byte. These fields are described below.

For cumulative log parameter values (indicated by the PC field of the LOG SELECT and LOG SENSE command descriptor block), the disable update (DU) bit is defined as follows:

a) A zero value indicates that the device server shall update the log parameter value to reflect all events that should be noted by that parameter; or
b) A one value indicates that the device server shall not update the log parameter value except in response to a LOG SELECT command that specifies a new value for the parameter.

NOTE 44 When updating cumulative log parameter values, a device server may use volatile memory to hold these values until a LOG SELECT or LOG SENSE command is received with an SP bit of one (or a target-defined event occurs). Thus the updated cumulative log parameter values may be lost if a power cycle occurs.

The DU bit is not defined for threshold values (indicated by the PC field of the LOG SENSE command descriptor block) nor for list parameters (indicated by the LP bit). The device server shall ignore the value of any DU bits in a LOG SELECT command.

A disable save (DS) bit of zero indicates that the target supports saving for that log parameter. The device server shall save the current cumulative or the current threshold parameter value (depending on the value in the PC field of the command descriptor block) in response to a LOG SELECT or LOG SENSE command with an SP bit of one. A DS bit of one indicates that the target does not support saving that log parameter in response to a LOG SELECT or LOG SENSE command with an SP bit of one.

A target save disable (TSD) bit of zero indicates that the target provides a target-defined method for saving log parameters. This implicit saving operation shall be done frequently enough to insure that the cumulative parameter values retain statistical significance (i.e., across power cycles). A TSD bit of one indicates that either the target does not provide a target-defined method for saving log parameters or the target-defined method has been disabled individually by an application client setting the TSD bit to one. An application client may disable the target-defined method for saving all log parameters without changing any TSD bits. See the GLTSD bit in the control mode page (see 8.3.4).

An enable threshold comparison (ETC) bit of one indicates that a comparison to the threshold value is performed whenever the cumulative value is updated. An ETC bit of zero indicates that a comparison is not performed. The value of the ETC bit is the same for cumulative and threshold parameters.

The threshold met criteria (TMC) field (see table 83) defines the basis for comparison of the cumulative and threshold values. The TMC field is valid only if the ETC bit is one. The value of the TMC field is the same for cumulative and threshold parameters.

Table 83 — Threshold met criteria

Code	Basis for comparison
00b	Every update of the cumulative value
01b	Cumulative value equal threshold value
10b	Cumulative value not equal threshold value
11b	Cumulative value greater than threshold value

If the ETC bit is one and the result of the comparison is true, a unit attention condition shall be generated for all initiators. When reporting the unit attention condition, the device server shall set the sense key to UNIT ATTENTION and set the additional sense code to THRESHOLD CONDITION MET.

The LBIN bit is only valid if the LP is one. If the LP bit is one and the LBIN bit is zero then the list parameter is a string of ASCII graphic codes (i.e., code values 20h through 7Eh). If the LP bit is one and the LBIN bit is one then the list parameter is a list of binary information.

The list parameter (LP) bit indicates the format of the log parameter. If an application client attempts to set the value of the LP bit to a value other than the one returned for the same parameter in the LOG SENSE command, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

An LP bit of zero indicates that the parameter is a data counter. Data counters are associated with one or more events; the data counter is updated whenever one of these events occurs by incrementing the counter value. If each data counter has associated with it a target-defined maximum value. Upon reaching this maximum value, the data counter shall not be incremented (i.e., it does not wrap). When a data counter reaches its maximum value, the device server shall set the associated DU bit to one. If the data counter is at or reaches its maximum value during the execution of a command, the device server shall complete the command. If the command completes correctly (except for the data counter being at its maximum value) and if the RLEC bit of the control mode page (8.3.1) is set to one; then the device server shall terminate the command with CHECK CONDITION status and set the sense key to RECOVERED ERROR with the additional sense code set to LOG COUNTER AT MAXIMUM.

An LP bit of one indicates that the parameter is a list parameter. List parameters are not counters and thus the ETC and TMC fields shall be set to zero.

If more than one list parameter is defined in a single log page, the following rules apply to assigning parameter codes:

a) The parameter updated last shall have a higher parameter code than the previous parameter, except as defined in rule b);
b) When the maximum parameter code value supported by the target is reached, the device server shall assign the lowest parameter code value to the next log parameter (i.e., wrap-around parameter codes). If the associated command completes correctly (except for the parameter code being at its maximum value) and if the RLEC bit of the control mode page (8.3.1) is set to one; then the device server shall terminate the command with CHECK CONDITION status and set the sense key to RECOVERED ERROR with the additional sense code set to LOG LIST CODES EXHAUSTED.

NOTE 45 List parameters may be used to store the locations of defective blocks in the following manner. When a defective block is identified, a list parameter is updated to reflect the location and cause of the defect. When the next defect is encountered, the list parameter with the next higher parameter code is updated to record this defect. The size of the page may be made vendor-specific to accommodate memory limitations. It is recommended that one or more data counter parameters be defined for the page to keep track of the number of valid list parameters and the parameter code of the parameter with the oldest recorded defect. This technique may be adapted to record other types of information.

The parameter length field specifies the length in bytes of the following parameter value. If the application client sends a parameter length value that results in the truncation of the parameter value, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

If the application client sends a log parameter value that is outside the range supported by the target, and rounding is implemented for that parameter, the device server may either:

a) round to an acceptable value and terminate the command as described in 5.2; or
b) terminate the command with CHECK CONDITION status and set the sense key to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

When any counter in a log page reaches its maximum value, incrementing of all counters in that log page shall cease until reinitialized by the application client via a LOG SELECT command. If the RLEC bit of the control mode page is one, then the device server shall report the exception condition.

The page code assignments for the log pages are listed in table 84.

Table 84 — Log page codes

Page code	Description	Clause
01h	Buffer over-run/under-run page	8.2.1
03h	Error counter page (read) page	8.2.2
04h	Error counter page (read reverse) page	8.2.2
05h	Error counter page (verify) page	8.2.2
02h	Error counter page (write) page	8.2.2
0Bh	Last *n* deferred errors or asynchronous events page	8.2.3
07h	Last *n* error events page	8.2.4
06h	Non-medium error page	8.2.5
00h	Supported log pages	8.2.6
08h - 0Ah	Reserved (may be used by specific device types)	
0Ch - 2Fh	Reserved (may be used by specific device types)	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

8.2.1 Buffer over-run/under-run page

The buffer over-run/under-run page (page code 01h) defines 24 data counters that may be used to record the number of buffer over-runs or under-runs for the logical unit. A target that implements this page may implement one or more of the defined data counters.

A buffer over-run or under-run may occur when an initiator does not transmit data to or from the target's buffer fast enough to keep up with reading or writing the media. The cause of this problem is protocol-specific. A buffer over-run condition may occur during a read operation when a buffer full condition prevents continued transfer of data from the media to the buffer. A buffer under-run condition may occur during a write operation when a buffer empty condition prevents continued transfer of data to the media from the buffer. Most devices incur a delay at this point while the media is repositioned.

Table 85 defines the parameter code field for the buffer over-run/under-run counters.

Table 85 − Parameter code field for buffer over-run/under-run counters

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Count basis			Cause				Type

The parameter code field for buffer over-run/under-run counters is a 16-bit value comprised of eight reserved bits, a three-bit count basis field (see table 86), a four-bit cause field (see table 87), and a one-bit type field. These are concatenated to determine the value of the parameter code for that log parameter. For example, a counter for parameter code value of 0023h specifies a count basis of 001b; a cause of 0001b; and a type of 1b; this counter is incremented once per command that experiences an over-run due to the SCSI bus being busy.

The count basis field defines the criteria for incrementing the counter. The criteria are defined in table 86.

Table 86 − Count basis definition

Count basis	Description
000b	Undefined
001b	Per command
010b	Per failed reconnect
011b	Per unit of time
100b - 111b	Reserved

NOTE 46 The per unit of time count basis is device type specific. Direct-access devices typically use a latency period (i.e., one revolution of the medium) as the unit of time.

The cause field indicates the reason that the over-run or under-run occurred. The following causes are defined in table 87.

Table 87 − Cause field definition

Cause	Description
0h	Undefined
1h	Bus busy
2h	Transfer rate too slow
3h - Fh	Reserved

The type field indicates whether the counter records under-runs or over-runs. A value of zero specifies a buffer under-run condition and a value of one specifies a buffer over-run condition.

The counters contain the total number of times buffer over-run or under-run conditions have occurred since the last time the counter was cleared. The counter shall be incremented for each occurrence of an under-run or over-run condition and may be incremented more than once for multiple occurrences during the execution of a single command.

8.2.2 Error counter pages

This clause defines the optional error counter pages for write errors (page code 02h), read errors (page code 03h), read reverse errors (page code 04h) and verify errors (page code 05h). The log page format is defined near the beginning of 8.2. A page may return one or more log parameters that record events defined by the parameter codes.

Table 88 defines the parameter codes for the error counter pages. Support of each log parameter is optional.

Table 88 — Parameter codes for error counter pages

Parameter code	Description
0000h	Errors corrected without substantial delay
0001h	Errors corrected with possible delays
0002h	Total (e.g., rewrites or rereads)
0003h	Total errors corrected
0004h	Total times correction algorithm processed
0005h	Total bytes processed
0006h	Total uncorrected errors
0007h - 7FFFh	Reserved
8000h - FFFFh	Vendor-specific

NOTE 47 The exact definition of the error counters is not part of this standard. These counters should not be used to compare products because the products may define errors differently.

8.2.3 Last *n* deferred errors or asynchronous events page

Log page (0Bh) provides for a number of deferred errors or asynchronous events sense data records using the list parameter format of the log page. The number of these deferred errors or asynchronous events records supported, *n*, is vendor-specific. Each deferred error or asynchronous event record contains SCSI sense data for a deferred error or asynchronous event that has occurred. The parameter code associated with the record indicates the relative time at which the deferred error or asynchronous event occurred. A higher parameter code indicates that the deferred error or asynchronous event occurred later in time.

The content of the parameter value field of each log parameter is the SCSI sense data describing the deferred error.

The fields DU, TSD, ETC, TMC are reserved and shall be set to zero. The LBIN bit shall be set to one (binary information). LP shall bit be set to one (list parameter).

8.2.4 Last *n* error events page

Log page (07h) provides for a number of error-event records using the list parameter format of the log page. The number of these error-event records supported, *n*, is vendor-specific. Each error-event record contains vendor-specific diagnostic information for a single error encountered by the device. The parameter code associated with error-event record indicates the relative time at which the error occurred. A higher parameter code indicates that the error event occurred later in time.

The content of the parameter value field of each log parameter is an ASCII character string which may describe the error event. The exact contents of the character string is not defined by this standard.

When the last supported parameter code is used by an error-event record, the recording on this page of all subsequent error information shall cease until one or more of the list parameters with the highest parameter codes have been reinitialized. If the RLEC bit of the control mode page (8.3.1) is set to one, the device server shall return CHECK CONDITION status with the sense key set to RECOVERED ERROR and the additional sense code set to LOG LIST CODES EXHAUSTED. Alternatively, the device server may report this condition via asynchronous event notification (see SAM).

8.2.5 Non-medium error page

This page (page code 06h) provides for summing the occurrences of recoverable error events other than write, read, or verify failures. No discrimination among the various types of events is provided by parameter code (see table 89). Vendor-specific discrimination may be provided through the vendor-specific parameter codes.

Table 89 − Non-medium error event parameter codes

Parameter code	Description
0000h	Non-medium error count
0001h - 7FFFh	Reserved
8000h - FFFFh	Vendor-specific error counts

8.2.6 Supported log pages

The supported log page (see table 90) returns the list of log pages implemented by the target. Targets that implement the LOG SENSE command shall implement this log page.

Table 90 − Supported log pages

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved		Page code (00h)					
1	Reserved							
2	(MSB)							
3	Page length (n-3)							(LSB)
4								
n	Supported page list							

This page is not defined for the LOG SELECT command. This log page returns the list of supported log pages for the specified logical unit.

The page length field specifies the length in bytes of the following supported page list.

The supported page list field shall contain a list of all log page codes implemented by the target in ascending order beginning with page code 00h.

8.3 Mode parameters

This clause describes the block descriptors and the pages used with MODE SELECT and MODE SENSE commands that are applicable to all SCSI devices. Pages specific to each device type are described in the command standard (see 3.1.11) that applies to that device type.

The mode parameter list shown in table 91 contains a header, followed by zero or more block descriptors, followed by zero or more variable-length pages. Parameter lists are defined for each device type.

Table 91 − Mode parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0 - n	Mode parameter header							
0 - n	Block descriptor(s)							
0 - n	Page(s)							

8.3.1 Mode parameter header formats

The six-byte command descriptor block parameter header is defined in table 92.

Table 92 − Mode parameter header(6)

Bit / Byte	7	6	5	4	3	2	1	0
0	Mode data length							
1	Medium type							
2	Device-specific parameter							
3	Block descriptor length							

The ten-byte command descriptor block parameter header is defined in table 93.

Table 93 − Mode parameter header(10)

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB) Mode data length							
1	(LSB)							
2	Medium type							
3	Device-specific parameter							
4	Reserved							
5	Reserved							
6	(MSB) Block descriptor length							
7	(LSB)							

When using the MODE SENSE command, the mode data length field specifies the length in bytes of the following data that is available to be transferred. The mode data length does not include itself. When using the MODE SELECT command, this field is reserved.

NOTE 48 Targets that support more than 256 bytes of block descriptors and pages may need to implement ten-byte mode commands. The mode data length field in the six-byte command descriptor block header limits the returned data to 256 bytes.

Medium types are unique for each device type. Refer to the mode parameters clause of the specific device type command standard (see 3.1.11) for definition of these values. Some device types reserve this field.

The device specific parameter is unique for each device type. Refer to the mode parameters clause of the specific device type command standard (see 3.1.11) for definition of this field. Some device types reserve all or part of this field.

The block descriptor length specifies the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight, and does not include pages or vendor-specific parameters, if any, that may follow the last block descriptor. A block descriptor length of zero indicates that no block descriptors are included in the mode parameter list. This condition shall not be considered an error.

8.3.2 Mode parameter block descriptor formats

8.3.2.1 General block descriptor format

The mode parameter block descriptor format for all device types except direct-access is shown in table 94.

Table 94 − General mode parameter block descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Density code							
1	(MSB)							
2	Number of blocks							
3								(LSB)
4	Reserved							
5	(MSB)							
6	Block length							
7								(LSB)

Block descriptors specify some of the medium characteristics for all or part of a logical unit. Support for block descriptors is optional. Each block descriptor contains a density code field, a number of blocks field, and a block length field. Block descriptor values are always current (i.e., saving is not supported). A unit attention condition (see 7.8 and SAM) shall be generated when any block descriptor values are changed.

The density code field is unique for each device type. Refer to the mode parameters clause of the specific device type command standard (see 3.1.11) for definition of this field. Some device types reserve all or part of this field.

The number of blocks field specifies the number of logical blocks on the medium to which the density code and block length fields apply. A value of zero indicates that all of the remaining logical blocks of the logical unit shall have the medium characteristics specified.

NOTES

49 There may be implicit association between parameters defined in the pages and block descriptors. In this case, the target may change parameters not explicitly sent with the MODE SELECT command. A subsequent MODE SENSE command may be used to detect these changes.

50 The number of remaining logical blocks may be unknown for some device types.

The block length specifies the length in bytes of each logical block described by the block descriptor. For sequential-access devices, a block length of zero indicates that the logical block size written to the medium is specified by the transfer length field in the command descriptor block (see SSC).

8.3.2.2 Direct-access device block descriptor format

The mode parameter block descriptor format for the direct-access device type is shown in table 95.

Table 95 − Direct-access device mode parameter block descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				Number of blocks				
2								
3								(LSB)
4				Density code				
5	(MSB)							
6				Block length				
7								(LSB)

This block descriptor format shall apply only to direct-access devices. All other device types shall use the block descriptor format described in 8.3.2.1.

Block descriptors specify some of the medium characteristics for a logical unit. Support for block descriptors is optional. Each block descriptor contains a density code field, a number of blocks field, and a block length field. A unit attention condition (see 7.8 and SAM) shall be generated when any block descriptor values are changed.

The number of blocks field specifies the number of logical blocks on the medium to which the density code and block length fields apply. A value of zero indicates that all of the remaining logical blocks of the logical unit shall have the medium characteristics specified.

If the SCSI device doesn't support changing its capacity by changing the number of blocks field (via a MODE SELECT command), the value in the number of blocks field is ignored. If the device supports changing its capacity by changing the number of blocks field, then the number of blocks field is interpreted as follows:

a) If the number of blocks is set to zero, the device shall retain its current capacity if the block size has not changed. If the number of blocks is set to zero and the block size has changed, the device shall be set to its maximum capacity when the new block size takes effect;

b) If the number of blocks is greater than zero and less than or equal to its maximum capacity, the device shall be set to that number of blocks. If the block size has not changed, the device shall not become format corrupted. This capacity setting shall be retained through reset events or power cycles;
c) If the number of blocks field is set to a value greater than the maximum capacity of the device and less than FFFFFFFFh, then the command is terminated with a CHECK CONDITION status. The sense key is set to ILLEGAL REQUEST. The device shall retain its previous block descriptor settings;
d) If the number of blocks is set to FFFFFFFFh, the device shall be set to its maximum capacity. If the block size has not changed, the device shall not become format corrupted. This capacity setting shall be retained through reset events or power cycles.

NOTE 51 There may be implicit association between parameters defined in the pages and block descriptor. For direct-access devices, the block length affects the optimum values (the value that achieves the best performance) for the sectors per track, bytes per physical sector, track skew factor, and cylinder skew factor fields in the format parameters page. In this case, the target may change parameters not explicitly sent with the MODE SELECT command. A subsequent MODE SENSE command may be used to detect these changes.

The density code field is unique for each device type. Refer to the mode parameters clause of the specific device type command standard (see 3.1.11) for the definition of this field. Some device types reserve all or part of this field.

The block length specifies the length in bytes of each logical block described by the block descriptor.

8.3.3 Mode page format

The mode page format is defined in table 96.

Table 96 − Mode page format

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code					
1	Page length (n-1))							
2	Mode parameters							
n								

Each mode page contains a page code, a page length, and a set of mode parameters. The page codes are defined in this clause and in the mode parameter clauses in the command standard (see 3.1.11) for the specific device type.

When using the MODE SENSE command, a parameters savable (PS) bit of one indicates that the mode page may be saved by the target in a non-volatile, vendor-specific location. A PS bit of zero indicates that the supported parameters cannot be saved. When using the MODE SELECT command, the PS bit is reserved.

The page code field identifies the format and parameters defined for that mode page. Some page codes are defined as applying to all device types and other page codes are defined for the specific device type. The page codes that apply to a specific device type are defined in the command standard (see 3.1.11) for that device type.

When using the MODE SENSE command, if page code 00h (vendor-specific page) is implemented, the device server shall return that page last in response to a request to return all pages (page code 3Fh). When using the MODE SELECT command, this page should be sent last.

The page length field specifies the length in bytes of the mode parameters that follow. If the application client does not set this value to the value that is returned for the page by the MODE SENSE command, the device server shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST. The target is permitted to implement a mode page that is less than the full page length defined in this standard, provided no field is truncated and the page length field correctly specifies the actual length implemented.

The mode parameters for each page are defined in the following clauses, or in the mode parameters clause in the command standard (see 3.1.11) for the specific device type. Mode parameters not implemented by the target shall be set to zero.

Table 97 defines the mode pages that are applicable to all device types that include the MODE SELECT and MODE SENSE commands.

Table 97 — Mode page codes

Page code	Description	Clause
0Ah	Control mode page	8.3.4
02h	Disconnect-reconnect page	8.3.5
1Ch	Informational exceptions control page	8.3.6
09h	obsolete	3.3.5
1Ah	Power condition page	8.3.7
01h	(See specific device type)	
03h - 08h	(See specific device type)	
0Bh - 19h	(See specific device type)	
1Bh	(See specific device type)	
1Dh - 1Fh	(See specific device type)	
00h	Vendor-specific (does not require page format)	
20h - 3Eh	(See specific device type)	
3Fh	Return all pages (valid only for the MODE SENSE command)	

8.3.4 Control mode page

The control mode page (see table 98) provides controls over several SCSI-3 features that are applicable to all device types such as tagged queuing, asynchronous event reporting, and error logging.

Table 98 – Control mode page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (0Ah)					
1	Page length (0Ah)							
2	Reserved						GLTSD	RLEC
3	Queue algorithm modifier				Reserved		QErr	DQue
4	Reserved	RAC	Reserved		SWP	RAERP	UAAERP	EAERP
5	Reserved							
6	(MSB) Ready AER holdoff period							
7	(LSB)							
8	(MSB) Busy timeout period							
9	(LSB)							
10	Reserved							
11	Reserved							

A global logging target save disable (GLTSD) bit of zero allows the target to provide a target-defined method for saving log parameters. A GLTSD bit of one indicates that either the target has disabled the target-defined method for saving log parameters or when set by the initiator specifies that the target-defined method shall be disabled.

A report log exception condition (RLEC) bit of one specifies that the device server shall report log exception conditions as described in 8.2. A RLEC bit of zero specifies that the device server shall not report log exception conditions.

The queue algorithm modifier field (see table 99) specifies restrictions on the algorithm used for reordering tasks having the SIMPLE task attribute.

Table 99 – Queue algorithm modifier

Value	Definition
0h	Restricted reordering
1h	Unrestricted reordering allowed
2h - 7h	Reserved
8h - Fh	Vendor-specific

A value of zero in this field specifies that the device server shall order the actual execution sequence of tasks having the SIMPLE task attribute such that data integrity is maintained for that initiator. This means that, if the transmission of new service delivery requests is halted at any time, the final value of all data observable on the medium shall have exactly the

same value as it would have if all the tasks had been given the ORDERED task attribute. The restricted reordering value shall be the default value.

A value of one in this field specifies that the device server may reorder the actual execution sequence of tasks having the SIMPLE task attribute in any manner. Any data integrity exposures related to task sequence order shall be explicitly handled by the application client through the selection of appropriate commands and task attributes.

A queue error management (QErr) bit of zero specifies that the blocked tasks in the task set shall resume after an ACA condition is cleared (see SAM).

A QErr bit of one specifies all the blocked tasks in the task set shall be aborted when the COMMAND TERMINATED or CHECK CONDITION status is sent. A unit attention condition (see SAM) shall be generated for each initiator that had blocked tasks aborted except for the initiator to which the COMMAND TERMINATED or CHECK CONDITION status was sent. The device server shall set the additional sense code to COMMANDS CLEARED BY ANOTHER INITIATOR.

A disable queuing (DQue) bit of zero specifies that tagged queuing shall be enabled if the device server supports tagged queuing. A DQue bit of one specifies that tagged queuing shall be disabled. Any queued commands received by the device server shall be aborted. The method used to abort queued commands is protocol-specific.

The report a check (RAC) bit provides control of reporting long busy conditions or CHECK CONDITION status. A RAC bit of one specifies that a CHECK CONDITION status should be reported rather than a long busy condition (e.g., longer than the Busy Timeout Period). A RAC bit of zero specifies that long busy conditions (e.g., busy condition during auto contingence allegiance) may be reported.

A software write protect (SWP) bit of one specifies that the logical unit shall inhibit writing to the medium after writing all cached or buffered write data, if any. When SWP is one, all commands requiring writes to the medium shall return CHECK CONDITION status and shall set the sense key to DATA PROTECT and the additional sense code to WRITE PROTECTED. When SWP is one and the device model defines a write protect (WP) bit in the Device-specific parameter in the Mode parameter header, the WP bit shall be set to one for subsequent MODE SENSE commands. A SWP bit of zero specifies that the logical unit may allow writing to the medium, depending on other write inhibit mechanisms implemented by the logical unit. When the SWP bit is zero, the value of the WP bit (if defined) is device model specific. For a list of commands affected by the SWP bit and details of the WP bit (if defined) see the command standard (see 3.1.11) for the specific device type.

The RAERP, UAAERP, and EAERP bits enable specific events to be reported via the asynchronous event reporting protocol. When all three bits are zero, the target shall not use asynchronous event reporting. AER is defined in SAM.

A ready AER permission (RAERP) bit of one specifies that the device server may issue an asynchronous event report upon completing its initialization sequence instead of generating a unit attention condition. A RAERP bit of zero specifies that the device server shall not issue an asynchronous event report upon completing its initialization sequence.

NOTE 52 If the device server's default value for the RAERP bit is one and it does not implement saved parameters or include a hardware switch, then it may be impossible to disable the initialization sequence asynchronous event reporting.

A unit attention AER permission (UAAERP) bit of one specifies that the device server may issue an asynchronous event report instead of creating a unit attention condition upon detecting an unit attention condition event (other than upon completing an initialization sequence). A UAAERP bit of zero specifies that the device server shall not issue an asynchronous event reporting instead of creating a unit attention condition.

An error AER permission (EAERP) bit of one specifies that the device server may issue an asynchronous event report upon detecting a deferred error condition instead of waiting to report the deferred error on the next command. An EAERP bit of zero specifies that the device server shall not report deferred error conditions via an asynchronous event reporting.

The ready AER holdoff period field specifies the minimum time in milliseconds after the target starts its initialization sequence that it shall delay before attempting to issue an asynchronous event report. This value may be rounded up as defined in the 5.2.

The busy timeout period field specifies the maximum time, in 100 milliseconds increments, that the initiator allows for the target to remain busy for unanticipated conditions which are not a routine part of commands from the initiator. This value may be rounded down as defined in 5.2. A 0000h value in this field is undefined by this standard. An FFFFh value in this field is defined as an unlimited period.

8.3.5 Disconnect-reconnect page

The disconnect-reconnect page (see table 100) provides the application client the means to tune the performance of the service delivery subsystem. The name for this mode page (disconnect-reconnect) comes from the SCSI-2 parallel bus. An SCSI-3 device based on any of the protocols may use appropriate parameters in the disconnect-reconnect mode page. The parameters appropriate to each protocol and their interpretation for that protocol may be specified in the individual protocol documents.

Table 100 – Disconnect-reconnect page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (02h)					
1	Page length (0Eh)							
2	Buffer full ratio							
3	Buffer empty ratio							
4	(MSB)			Bus inactivity limit				
5								(LSB)
6	(MSB)			Disconnect time limit				
7								(LSB)
8	(MSB)			Connect time limit				
9								(LSB)
10	(MSB)			Maximum burst size				
11								(LSB)
12	EMDP	FARd	FAWrt	FAStat	DImm	DTDC		
13	Reserved							
14	(MSB)			First burst size				
15								(LSB)

The device server communicates the parameter values in this mode page to the service delivery subsystem (e.g., to its Target Role Agent). Similarly the application client may also communicate parameter values to the service delivery subsystem (e.g., those controlling behavior of its Initiator Role Agent). This communication is internal to the initiator or target device and is outside the scope of SCSI-3.

If a parameter that is not appropriate for the specific protocol implemented by the SCSI-3 device is non-zero, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

An interconnect tenancy is a period of time during which a target device owns or may access the interconnect. For example, on arbitrated interconnects, a tenancy typically begins when a device successfully arbitrates for the interconnect and ends when the device releases the interconnect for use by other devices. Data and other information transfers take place during interconnect tenancies.

The buffer full ratio field indicates to the device server, during read operations, how full the buffer should be prior to requesting an interconnect tenancy. Device servers that do not implement the requested ratio should round down to the nearest implemented ratio as defined in 5.2.

The buffer empty ratio field indicates to the device server, during write operations, how empty the buffer should be prior to requesting an interconnect tenancy. Device servers that do not implement the requested ratio should round down to the nearest implemented ratio as defined in 5.2.

The buffer full and buffer empty ratios are numerators of a fractional multiplier that has 256 as its denominator. A value of zero indicates that the target determines when to request an interconnect tenancy consistent with the disconnect time limit parameter. These parameters are advisory to the target.

NOTE 53 As an example, consider a device server with ten 512-byte buffers and a specified buffer full ratio of 3Fh. The formula is: INTEGER((ratio/256)*number of buffers). Thus INTEGER((3Fh/256)*10) = 2. During the read operations described in this example, the device server should request an interconnect tenancy whenever two or more buffers are full.

The bus inactivity limit field indicates the maximum time that the target is permitted to maintain an interconnect tenancy without data or information transfer. If the bus inactivity limit is exceeded the device server shall conclude the interconnect tenancy, within the restrictions placed on it by the applicable SCSI-3 protocol. The contents of the DTDC parameter in this mode page also shall affect the duration of an interconnect tenancy. This value may be rounded as defined in 5.2. A value of zero indicates that there is no bus inactivity limit.

The disconnect time limit field indicates the minimum time that the target shall wait between interconnect tenancies. This value may be rounded as defined in 5.2. A value of zero indicates that there is no disconnect time limit.

The connect time limit field indicates the maximum duration of a single interconnect tenancy. If the connect time limit is exceeded the device server shall conclude the interconnect tenancy, within the restrictions placed on it by the applicable SCSI-3 protocol. The contents of the DTDC parameter in this mode page also shall affect the duration of an interconnect tenancy. This value may be rounded as defined in 5.2. A value of zero indicates that there is no connect time limit.

The maximum burst size field indicates the maximum amount of data that the device server shall transfer during a single data transfer operation. This value is expressed in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1024 bytes, etc.). The relationship (if any) between data transfer operations and interconnect tenancies is specified in the individual protocol documents. A value of zero indicates there is no limit on the amount of data transferred per data transfer operation.

The enable modify data pointers (EMDP) bit indicates whether or not the initiator allows the data transfer to be re-ordered by the target. If the EMDP bit is zero, the target shall not re-order the data transfer. If the EMDP bit is one, the target is allowed to re-order the data transfer.

The Fair Arbitration Read (FARd), Fair Arbitration Write (FAWrt), and Fair Arbitration Status (FAStat) bits indicate whether the target should use fair or unfair (e.g., priority) arbitration when requesting an interconnect tenancy for a read data transfer, write data transfer, and status or control message transfer respectively. An FA bit of one indicates that the target should use fair arbitration. An FA bit of zero indicates that the target should use unfair (e.g., priority) arbitration.

A disconnect immediate (DImm) bit of zero indicates that the target may transfer data for a command during the same interconnect tenancy in which it receives the command. Whether or not the target does so may depend upon the target's internal algorithms, the rules of the applicable SCSI-3 protocol, and settings of the other parameters in this mode page. A disconnect immediate (DImm) bit of one indicates that the target shall not transfer data for a command during the same interconnect tenancy in which it receives the command.

The data transfer disconnect control (DTDC) field (see table 101) defines other restrictions on when multiple interconnect tenancies are permitted. A non-zero value in the DTDC field shall take precedence over other interconnect tenancy controls represented by other fields in this mode page.

Table 101 – Data transfer disconnect control

DTDC	Description
000b	Data transfer disconnect control is not used. Interconnect tenancies are controlled by other fields in this page.
001b	A target shall transfer all data for a command within a single interconnect tenancy.
010b	Reserved
011b	A target shall transfer all data for a command and complete the command within a single interconnect tenancy.
100b-111b	Reserved

The first burst size field indicates the maximum amount of data that a target may transfer for a command during the same interconnect tenancy in which it receives the command. This value is expressed in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1024 bytes, etc.). A value of zero indicates that there is no first burst size limit.

8.3.6 Informational exceptions control page

The informational exceptions control page (see table 102) defines the methods used by the target to control the reporting and the operations of specific informational exception conditions. This page shall only apply to informational exceptions that report an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the application client.

Informational exception conditions occur as the result of vendor-specific events within a target. An informational exception condition may occur asynchronous to any commands issued by an application client.

Table 102 — Informational exceptions control page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (1Ch)					
1	Page length (0Ah)							
2	Perf	Reserved			DExcpt	Test	Reserved	LogErr
3	Reserved				MRIE			
4	(MSB)							
7	Interval timer							(LSB)
8	(MSB)							
11	Report count							(LSB)

The log errors bit (LogErr) of zero indicates that the logging of informational exception conditions by a device server is vendor-specific. A LogErr bit of one indicates the device server shall log informational exception conditions.

A disable exception control (DExcpt) bit of zero indicates information exception operations shall be enabled. The reporting of information exception conditions when the DExcpt bit is set to zero is determined from the method of reporting informational exceptions field. A DExcpt bit of one indicates the device server shall disable all information exception operations. The method of reporting informational exceptions field is ignored when DExcpt is set to one.

A Test bit of one shall create a test device failure at the next interval time (as specified by the Interval timer field), if the DExcpt bit is not set. When the Test bit is one, the MRIE and Report count fields shall apply as if the Test bit were zero. The test device failure shall be reported with an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE). If both the Test and the DExcpt bits are one, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST. A Test bit of zero shall instruct the device server not to generate any test device failure notifications.

A Performance bit (Perf) of zero indicates that informational exception operations that are the cause of delays are acceptable. A Perf bit of one indicates the device server shall not cause delays while doing informational exception operations. A Perf bit set to one may cause the device server to disable some or all of the informational exceptions operations, thereby limiting the reporting of informational exception conditions.

The Method of Reporting Informational Exceptions field (MRIE) indicates the methods that shall be used by the device server to report informational exception conditions (see table 103). The priority of reporting multiple information exceptions is vendor-specific.

Table 103 — Method of Reporting Informational Exceptions field

MRIE	Description
0h	**No reporting of informational exception condition:** This method instructs the device server to not report information exception conditions.
1h	**Asynchronous event reporting:** This method instructs the device server to report informational exception conditions by using the rules for asynchronous event reporting as described in the SCSI-3 Architecture Model and the relevant protocol standard. The sense key shall be set to RECOVERED ERROR and the additional sense code shall indicate the cause of the informational exception condition.
2h	**Generate unit attention:** This method instructs the device server to report informational exception conditions by returning a CHECK CONDITION status. The sense key shall be set to UNIT ATTENTION and the additional sense code shall indicate the cause of the informational exception condition. The command that has the CHECK CONDITION shall not be executed before the informational exception condition is reported.
3h	**Conditionally generate recovered error:** This method instructs the device server to report informational exception conditions, if the reporting of recovered errors is allowed, by returning a CHECK CONDITION status. The sense key shall be set to RECOVERED ERROR and the additional sense code shall indicate the cause of the informational exception condition. A command that has the CHECK CONDITION shall complete without error before any informational exception condition may be reported.
4h	**Unconditionally generate recovered error:** This method instructs the device server to report informational exception conditions, regardless of the value of the per bit of the error recovery parameters mode page, by returning a CHECK CONDITION status on any command. The sense key shall be set to RECOVERED ERROR and the additional sense code shall indicate the cause of the informational exception condition. The command that has the CHECK CONDITION shall complete without error before any informational exception condition may be reported.
5h	**Generate no sense:** This method instructs the device server to report informational exception conditions by returning a CHECK CONDITION status. The sense key shall be set to NO SENSE and the additional sense code shall indicate the cause of the informational exception condition. The command that has the CHECK CONDITION shall complete without error before any informational exception condition may be reported.

(continued)

Table 103 — Method of Reporting Informational Exceptions field (concluded)

MRIE	Description
6h	**Only report informational exception condition on request:** This method instructs the device server to preserve the informational exception(s) information. To find out about information exception conditions the application client polls the device server by issuing an unsolicited REQUEST SENSE command. The sense key shall be set to NO SENSE and the additional sense code shall indicate the cause of the informational exception condition.
7h-Bh	Reserved
Ch-Fh	Vendor-specific

The Interval Timer field indicates the period in 100 millisecond increments for reporting that a informational exception condition has occurred. The device server shall not report informational exception conditions more frequently than the time specified by the Interval Timer field and as soon as possible after the timer interval has elapsed. After the informational exception condition has been reported the interval timer shall be restarted. A value of zero or FFFFFFFFh in the Interval Timer field shall indicate the timer interval is vendor-specific.

The Report Count field indicates the number of times to report an informational exception condition to the application client. A value of zero in the Report Count field indicates there is no limit on the number of times the device server reports an informational exception condition.

The maintaining of the Interval Timer and the Report Count fields across power cycles and/or resets by the target are vendor-specific.

8.3.7 Power condition page

The power condition page (see table 104) provides the application client the means to control the behavior of a logical unit in a manner which reduces the power required to operate. There shall be no notification to the initiator that a logical unit has entered into one of the power conditions. The application client may determine if a power condition is in effect by issuing a REQUEST SENSE command (see 7.20). In addition to the power condition page, the power conditions may be controlled by the START STOP UNIT command (see SBC). If both methods are being used on the same logical unit then any START STOP UNIT commands power condition request shall override the power condition pages power control.

No power condition shall affect the supply of any power required for proper operation of the service delivery subsystem.

On the receipt of a command the device server 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.

Logical units that contain cache memory shall implicitly perform a SYNCHRONIZE CACHE command (see SBC) for the entire medium prior to entering into any power condition which prevents access the media (e.g., 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 that sets power conditions.

Table 104 — Power condition page

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

An Idle bit of one indicates that the 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 that the logical unit shall not enter the Idle condition.

A Standby bit of one indicates that the 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 that the 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.

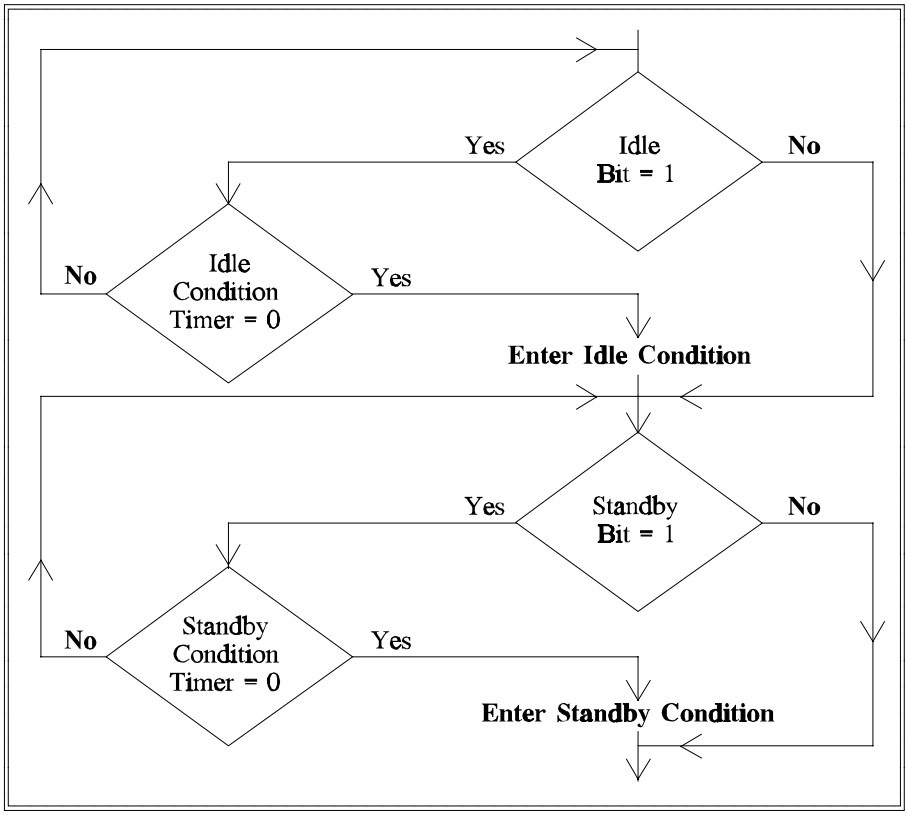

Figure 2 – Power conditions flowchart

8.4 Vital product data parameters

This clause describes the optional vital product data page structure and the vital product data pages (see table 105) that are applicable to all SCSI devices. These pages are optionally returned by the INQUIRY command (7.5) and contain vendor-specific product information about a target or logical unit. The vital product data may include vendor identification, product identification, unit serial numbers, device operating definitions, manufacturing data, field replaceable unit information, and other vendor-specific information. This standard defines the structure of the vital product data, but not the contents.

Table 105 – Vital product data page codes

Page code	Description	Clause
82h	ASCII implemented operating definition page	8.4.1
01h - 7Fh	ASCII information page	8.4.2
83h	Device identification page	8.4.3
81h	Implemented operating definitions page	8.4.4
00h	Supported vital product data pages	8.4.5
80h	Unit serial number page	8.4.6
84h - BFh	Reserved	
C0h - FFh	Vendor-specific	

If a device server supports any vital product data pages, it also shall support vital product data page code 00h.

8.4.1 ASCII implemented operating definition page

The ASCII implemented operation definition page (see table 106) contains operating definition description data for all operating definitions implemented by the target.

Table 106 – ASCII implemented operating definition

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Page code (82h)							
2	Reserved							
3	Page length (n-3)							
4	ASCII operating definition description length (m-4)							
5 – – – – m	ASCII operating definition description data – –							
m+1 – – – – n	Vendor-specific description data – –							

The peripheral qualifier field and the peripheral device type field are as defined in 7.5.1.

The page length field specifies the length of the following page data. If the allocation length is less than the length of the data to be returned, the page length shall not be adjusted to reflect the truncation.

The ASCII operating definition description length field specifies the length in bytes of the ASCII operating definition description data that follows. If the allocation length is less than the length of data to be returned, the ASCII operating definition description length shall not be adjusted to reflect the truncation. A value of zero in this field indicates that no ASCII operating definition description data is available.

The ASCII operating definition description data field contains the ASCII operating definition description data for the device server. The data in this field shall be formatted in lines (or character strings). Each line shall contain only graphic codes (i.e., code values 20h through 7Eh) and shall be terminated with a NULL (00h) character. The text is vendor-specific.

8.4.2 ASCII information page

The ASCII information page (see table 107) contains information for the field replaceable unit code returned in the REQUEST SENSE data (see 7.20).

Table 107 — ASCII information page

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Page code (01h - 7Fh)							
2	Reserved							
3	Page length (n-3)							
4	ASCII length (m-4)							
5 — m	ASCII information							
m+1 — n	Vendor-specific information							

The peripheral qualifier field and the peripheral device type field are defined in 7.5.1.

The page code field contains the same value as in the page code field of the INQUIRY command descriptor block (see 7.5) and is associated with the field replaceable unit code returned by the REQUEST SENSE command.

NOTE 54 The field replaceable unit field in the sense data provides for 255 possible codes, while the page code field provides for only 127 possible codes. Thus it is not possible to return ASCII information pages for the upper code values.

The page length field specifies the length of the following page data. If the allocation length of the command descriptor block is too small to transfer all of the page, the page length shall not be adjusted to reflect the truncation.

The ASCII length field specifies the length in bytes of the ASCII information that follows. If the allocation length is less than the length of the data to be returned, the ASCII length shall not be adjusted to reflect the truncation. A value of zero in this field indicates that no ASCII information is available for the specified page code.

The ASCII information field contains ASCII information concerning the field replaceable unit identified by the page code. The data in this field shall be formatted in one or more lines (or character strings). Each line shall contain only graphic codes (i.e., code values 20h through 7Eh) and shall be terminated with a NULL (00h) character.

The contents of the vendor-specific information field is not defined in this standard.

8.4.3 Device identification page

The device identification page (see table 108) provides the means to retrieve zero or more identification descriptors applying to the logical unit. Logical units may have more than one identification descriptor (e.g., if several types of identifier are supported).

Device identifiers, if any, shall be assigned to the peripheral device (e.g., a disk drive) and not to the currently mounted media, in the case of removable media devices. Media identification is outside the scope of this standard. Operating systems are expected to use the device identifiers during system configuration activities to determine whether alternate paths exist for the same peripheral device.

NOTE 55 In the case of virtual logical units (e.g., volume sets as defined by SCC), the Identifier field (see table 109) may be constructed in a vendor-specific manner. Vendors should ensure that such identifiers are globally unique and have an Identifier type value of 0.

Table 108 — Device identification page

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Page code (83h)							
2	Reserved							
3	Page length (n-3)							
	Identification descriptor list							
4	(MSB)							
	Identification descriptor (0) (see table 109)							(LSB)
	. . .							
n	(MSB)							
	Identification descriptor (last) (see table 109)							(LSB)

Table 109 — Identification descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved				Code set			
1	Reserved				Identifier type			
2	Reserved							
3	Identifier length (n-3)							
4	(MSB)							
	Identifier							
n								(LSB)

The peripheral qualifier field and the peripheral device type field in table 108 are as defined in 7.5.1.

Each Identification descriptor (see table 109) contains information identifying the logical unit. If the logical unit is accessible through any other path, it shall return the same identification.

The Code set field specifies the code set used for the identifier field, as described in table 110. This field is intended to be an aid to software that displays the identifier field.

Table 110 — Code set

Value	Description
0h	Reserved
1h	The identifier field shall contain binary values.
2h	The identifier field shall contain ASCII graphic codes (i.e., code values 20h through 7Eh)
3-Fh	Reserved

The Identifier type field specifies the format and assignment authority for the identifier, as described in table 111.

Table 111 — Identifier type

Value	Description
0	No assignment authority was used and consequently there is no guarantee that the identifier is globally unique (i.e., the identifier is vendor-specific)
1	The first 8 bytes of the identifier field are a Vendor ID (see annex C). The organization associated with the Vendor ID is responsible for ensuring that the remainder of the identifier field is unique. One recommended method of constructing the remainder of the identifier field is to concatenate the product identification field from the standard INQUIRY data field and the product serial number field from the unit serial number page.
2	The identifier field contains an IEEE Extended Unique Identifier, 64-bit (EUI-64). In this case, the identifier length field shall be set to 8. Note that the IEEE guidelines for EUI-64 specify a method for unambiguously encapsulating an IEEE 48-bit identifier within an EUI-64.
3	The identifier field contains a FC-PH 64-bit Name_Identifier field as defined in X3.230-1994. In this case, the identifier length shall be set to 8.
4-Fh	Reserved

The Identifier length field specifies the length in bytes of the Identifier. If the allocation length field of the command descriptor block is too small to transfer all of the identifier, the Identifier length shall not be adjusted to reflect the truncation.

The Identifier field contains the identifier as described by the Identifier type, Code set, and Identifier length fields.

The example described in this paragraph and shown in table 112 is not a normative part of this standard. This example of a complete device identification VPD page assumes that the product is a direct-access device with an T10 Vendor ID of "XYZ_Corp", a product identification of "Super Turbo Disk", and a product serial number of "2034589345". Furthermore, it is assumed that the manufacturer has been assigned a 24-bit IEEE company_id of 01ABCDh by the IEEE Registration Authority Committee and that the manufacture has assigned a 24-bit extension_identifier of 234567h to this logical unit. The combined 48-bit identifier is reported in the 64-bit format as defined by the IEEE 64-bit Global Identifier (EUI-64) standard. The data returned in the device identification VPD page for this logical unit is:

Table 112 — Device identification page example

Bytes	Hexadecimal Values	ASCII Values
00--15	00 83 00 32 02 01 00 22 58 59 5A 5F 43 6F 72 70	...2..."XYZ_Corp
16--31	53 75 70 65 72 20 54 75 72 62 6F 20 44 69 73 6B	Super Turbo Disk
32--47	32 30 33 34 35 38 39 33 34 35 01 02 00 08 01 AB	2034589345......
48--53	CD FF FF 23 45 67	......

Notes: a) Non-printing ASCII characters are shown as '.'.
b) Byte 00 is the beginning of the VPD page (see table 108).
c) Byte 04 is the beginning of the Identification descriptor for the Vendor ID based identifier (Identifier type 1, see table 111)
d) Byte 42 is the beginning of the Identification descriptor for the EUI-64 identifier (Identifier type 2, see table 111)

8.4.4 Implemented operating definition page

The implemented operating definition page (see table 113) defines the current operating definition, the default operating definition, and the operating definitions implemented by the target. These operating definition values are defined in the CHANGE DEFINITION command (see 7.1).

Table 113 — Implemented operating definition page

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Page code (81h)							
2	Reserved							
3	Page length (n-3)							
4	Reserved	Current operating definition						
5	SavImp	Default operating definition						
6	SavImp	Supported operating definition list						
n	SavImp							

The peripheral qualifier field and the peripheral device type field are defined in 7.5.1.

The page length field specifies the length of the following operating definitions. If the allocation length of the command descriptor block is too small to transfer all of the page, the page length shall not be adjusted to reflect the truncation.

For each operating definition, there is an associated save implemented (SavImp) bit. A SavImp bit of zero indicates that the corresponding operating definition parameter cannot be saved. A SavImp bit of one indicates that the corresponding operating definition parameter may be saved.

All returned operating definitions use the codes defined in table 7. The current operating definition field returns the value of the present operating definition. If no operating definition is saved, the default operating definition field returns the value of the operating definition the target uses when power is applied. The supported operating definition list returns one or more operating definitions implemented by the target.

8.4.5 Supported vital product data pages

This contains a list of the vital product data page codes supported by the target or logical unit (see table 114). If a device server supports any vital product data pages, it also shall support this vital product data page.

Table 114 – Supported vital product data pages

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Page code (00h)							
2	Reserved							
3	Page length (n-3)							
4 – – – – n	Supported page list							

The peripheral qualifier field and the peripheral device type field are defined in 7.5.1.

The page length field specifies the length of the supported page list. If the allocation length is too small to transfer all of the page, the page length shall not be adjusted to reflect the truncation.

The supported page list field shall contain a list of all vital product data page codes (see 8.4) implemented for the target or logical unit in ascending order beginning with page code 00h.

8.4.6 Unit serial number page

This page (see table 115) provides a product serial number for the target or logical unit.

Table 115 – Unit serial number page

Bit / Byte	7	6	5	4	3	2	1	0
0	Peripheral qualifier			Peripheral device type				
1	Page code (80h)							
2	Reserved							
3	Page length (n-3)							
4 – n	Product serial number							

The peripheral qualifier field and the peripheral device type field are defined in 7.5.1.

The page length field specifies the length of the product serial number. If the allocation length is too small to transfer all of the page, the page length shall not be adjusted to reflect the truncation.

The product serial number field contains ASCII data that is vendor-assigned serial number. The least significant ASCII character of the serial number shall appear as the last byte in the Data-In Buffer. If the product serial number is not available, the device server shall return ASCII spaces (20h) in this field.

9 Commands for processor type devices

The commands for processor type devices shall be as listed in table 116.

Table 116 − Commands for processor devices

Command name	Operation code	Type	Clause
CHANGE DEFINITION	40h	O	7.1
COMPARE	39h	O	7.2
COPY	18h	O	7.3
COPY AND VERIFY	3Ah	O	7.4
INQUIRY	12h	M	7.5
LOG SELECT	4Ch	O	7.6
LOG SENSE	4Dh	O	7.7
PERSISTENT RESERVE IN	5Eh	O	7.12
PERSISTENT RESERVE OUT	5Fh	O	7.13
READ BUFFER	3Ch	O	7.14
RECEIVE	08h	O	9.1
RECEIVE DIAGNOSTIC RESULTS	1Ch	O	7.16
RELEASE(6)	17h	O	7.17
RELEASE(10)	57h	O	7.18
REPORT LUNS	A0h	O	7.19
REQUEST SENSE	03h	M	7.20
RESERVE(6)	16h	O	7.21
RESERVE(10)	56h	O	7.22
SEND	0Ah	M	9.2
SEND DIAGNOSTIC	1Dh	M	7.23
TEST UNIT READY	00h	M	7.24
WRITE BUFFER	3Bh	O	7.25

Key: M = Command implementation is mandatory.
O = Command implementation is optional.

The following operation codes are vendor-specific: 02h, 05h, 06h, 09h, 0Ch, 0Dh, 0Eh, 0Fh, 10h, 11h, 13h, 14h, 19h, C0h through FFh. All remaining operation codes for processor devices are reserved.

9.1 RECEIVE command

The RECEIVE command (see table 117) requests that the device server transfer data to the initiator. The contents of the data are not defined by this standard.

Table 117 − RECEIVE command

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

If reservations are active, they shall affect the execution of the RECEIVE command as follows. A reservation conflict shall occur when a RECEIVE command is received from an initiator other than the one holding a logical unit reservation.

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

9.2 SEND command

The SEND command (see table 118) requests that the device server transfer data from the initiator.

Table 118 − SEND command

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

If reservations are active, they shall affect the execution of the SEND command as follows. A reservation conflict shall occur when a SEND command is received from an initiator other than the one holding a logical unit reservation.

An asynchronous event reporting (AER) bit of one indicates that the data to be transferred conforms to AER data format as defined in table 119. A SEND command with an AER bit of one shall be only issued to logical unit zero. An AER bit of zero indicates that the data to be transferred are vendor-specific.

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

Table 119 — SEND command - AER data format

Bit / Byte	7	6	5	4	3	2	1	0
0	SCSI-3	Reserved						
1 — 3	Reserved							
4 — 11	LUN							
12 — n+12	Sense data byte (0) — Sense data byte (n)							

If the SCSI-3 bit is zero, then the AEN data format (as defined by the SCSI-2 standard) shall be used. If the SCSI-3 bit is one, then the AER data format shown in table 119 shall be used.

The LUN field shall contain the logical unit number on which the asynchronous event occurred. The LUN field shall have the properties defined in SAM.

The sense data bytes shall have the format defined in 7.20.

10 Parameters for processor type devices

10.1 Diagnostic parameters

This clause defines the descriptors and pages for diagnostic parameters used with processor type devices.

The diagnostic page codes for processor devices are defined in table 120.

Table 120 — Processor diagnostic page codes

Page code	Description	Clause
00h	Supported diagnostics pages	8.1.1
01h - 3Fh	Reserved (for uses that apply to all device types)	
40h - 7Fh	Reserved	
80h - FFh	Vendor-specific pages	

10.2 Log parameters

This clause defines the descriptors and pages for log parameters used with processor type devices.

The log page codes for processor devices are defined in table 121.

Table 121 — Processor log page codes

Page code	Description	Clause
01h	Buffer over-run/under-run page	8.2.1
0Bh	Last *n* deferred errors or asynchronous events page	8.2.3
07h	Last *n* error events page	8.2.4
06h	Non-medium error page	8.2.5
00h	Supported log pages	8.2.6
02h - 05h	Reserved	
08h - 0Ah	Reserved	
0Bh - 2Fh	Reserved	
3Fh	Reserved	
30h - 3Eh	Vendor-specific pages	

Annex A
(informative)

Procedures for logging operations in SCSI

This annex provides guidance in the use of the LOG SELECT and LOG SENSE commands defined in clause 7. This annex does not replace the descriptions in clause 7 and is not intended to conflict with clause 7. The purpose of this annex is to provide more information to gain a more uniform implementation of the SCSI logging functions.

A.1 Logging operations terminology

A.1.1 list parameter: A parameter value that consists of a string of ASCII graphic codes or a binary value.

A.1.2 log page: A page made up of one or more log parameters.

A.1.3 log parameter: Log information that is made up of a parameter code, a parameter control byte, and a parameter value.

A.1.4 parameter code: A unique identifier that is used to distinguish between the different log parameters within a single log page.

A.1.5 parameter control byte: Used to tell the device server how to update, save, use thresholds, determine format, etc. of the parameter value.

A.1.6 parameter pointer field: Contains a parameter code.

A.1.7 parameter value: A counter, cumulative, threshold, or ASCII value.

A.1.8 nv: Not Valid

A.1.9 x: The value of the bit or field is not relevant.

A.2 LOG SENSE command

The LOG SENSE command may be used to do two functions. One is to allow the device server to save the log parameters in a log page to non-volatile storage. The other is to allow an application client to receive the value of the current log parameters for a given log page.

Table A.1 lists the definitions of the LOG SENSE Command Descriptor Block (CDB) fields:

Table A.1 – LOG SENSE Command CDB fields

LOG SENSE CDB Values			
PPC bit	SP bit	PC field	**Description**
0	-	--	Indicates that the log parameter requested from the device server begin with the parameter code specified by the parameter pointer field in ascending order of parameter codes from the specified log page.
1	-	--	Indicates that the device server returns a log page consisting only of the log parameters in which a log parameter value has changed since the last LOG SELECT or LOG SENSE command. The device server returns only those log parameters following the parameter pointer field.
-	0	--	Indicates that the device server performs the specified LOG SENSE command and does not save any log parameters.
-	1	--	Indicates that the device server performs the specified LOG SENSE command and saves all log parameters identified as savable by the DS bit to a non-volatile vendor-specific location if allowed. (See the Table A.3 to determine the interaction between the SP and DS bits to see what 'allowed' means.)
-	-	00	Indicates that the device server returns current threshold values.
-	-	01	Indicates that the device server returns current cumulative values.
-	-	10	Indicates that the device server returns default threshold values.
-	-	11	Indicates that the device server returns default cumulative values.

Table A.2 lists all possible parameter values that may be returned by a LOG SENSE command.

Table A.2 – LOG SENSE returned parameter values

LOG SENSE CDB Values		Log Page Parameter Control Byte Value		Device Server Action
PPC bit	PC field	LP bit	LBIN bit	Parameter values returned to the application client
0	00	0	x	Returns all current threshold values starting with the specified parameter pointer.
0	01	0	x	Returns all current cumulative values starting with the specified parameter pointer.
0	10	0	x	Returns all default threshold values starting with the specified parameter pointer.
0	11	0	x	Returns all default cumulative values starting with the specified parameter pointer.
1	00	0	x	Returns only the current threshold values that have changed, starting with the specified parameter pointer.
1	01	0	x	Returns only the current cumulative values that have changed, starting with the specified parameter pointer.
1	10	0	x	Returns only the default threshold values that have changed, starting with the specified parameter pointer.
1	11	0	x	Returns only the default cumulative values that have changed, starting with the specified parameter pointer.
0	xx	1	0	Returns all the list parameters starting with the specified parameter pointer. The list parameters returned are formatted as ASCII graphic codes.
1	xx	1	0	Returns only the list parameters that have changed, starting with the specified parameter pointer. The list parameters returned are formatted as ASCII graphic codes.
0	xx	1	1	Returns all the list parameters starting with the specified parameter pointer. The list parameters returned are formatted in binary.
1	xx	1	1	Returns only the list parameters that have changed, starting with the specified parameter pointer. The list parameters returned are formatted in binary.

Table A.3 lists all possible save options for the LOG SENSE command.

The listed options define the save operations that occur as a direct result of the LOG SENSE command. Further save operations are a function of the TSD bit in the log parameter control byte.

Table A.3 – LOG SENSE save options

LOG SENSE CDB Values		**Log Page Parameter Control Byte Value**			
SP bit	PC field	DS bit	LP bit	LBIN bit	**Device server action**
0	xx	x	x	x	Do not save any of the log parameters into non-volatile storage.
1	00	0	0	x	Save all the current threshold values of the selected log page into non-volatile storage.
1	01	0	0	x	Save all the current cumulative values of the selected log page into non-volatile storage.
1	10	0	0	x	Save all the default threshold values of the selected log page into non-volatile storage.
1	11	0	0	x	Save all the default cumulative values of the selected log page into non-volatile storage.
1	xx	0	1	0	Save all the current list parameter values of the selected log page into non-volatile storage. The list parameters are formatted as ASCII graphic codes.
1	xx	0	1	1	Save all the current list parameter values of the selected log page into non-volatile storage. The list parameters are formatted in binary.
1	xx	1	x	x	Do not save any of the log parameters into non-volatile storage.

A.3 LOG SELECT command

The function of the LOG SELECT command is to allow an application client a method of sending parameter values to the device server.

Table A.4 lists the definitions of the LOG SELECT Command Descriptor Block (CDB) fields.

Table A.4 – LOG SELECT CDB fields

LOG SELECT CDB Values				
PCR bit	SP bit	PC field	Parameter List Length	**Description**
0	-	--	-	Indicates that the log parameters are not reset.
1	x	xx	0000h	Indicates that the device server sets all implemented parameter values to the target-defined default values.
1	x	xx	GT 0	This is an illegal condition.
-	0	--	-	Indicates that the device server does not save any of the log parameters.
-	1	--	-	Indicates that, after performing the specified LOG SELECT operation, the device server saves to non-volatile memory all savable log parameters. (See table A.3 to determine the interaction between the SP and DS bits to see what 'savable' means.)
-	-	00	-	Indicates that the application client sends threshold values.
-	-	01	-	Indicates that the application client sends cumulative values.
-	-	10	-	Indicates that the application client sends default threshold values.
-	-	11	-	Indicates that the application client sends default cumulative values.

Table A.5 lists all possible save options for the LOG SELECT command.

All the Log Parameters that are selected for saving are saved to non-volatile storage after the device server performs the specified LOG SELECT operation. Further save operations are a function of the TSD bit in the log parameter control byte.

Table A.5 – LOG SELECT save options

LOG SELECT CDB Values		**Log Page Parameter Control Byte Value**			
SP bit	PC field	DS bit	LP bit	LBIN bit	**Device server action**
0	xx	x	x	x	Do not save any of the log parameters into non-volatile storage.
1	00	0	0	x	Save all the threshold values of the selected log page into non-volatile storage.
1	01	0	0	x	Save all the cumulative values of the selected log page into non-volatile storage.
1	10	0	0	x	Save all the default threshold values of the selected log page into non-volatile storage.
1	11	0	0	x	Save all the default cumulative values of the selected log page into non-volatile storage.
1	xx	0	1	0	Save all the list parameter values of the selected log page into non-volatile storage. The list parameters are formatted as ASCII graphic codes.
1	xx	0	1	1	Save all the list parameter values of the selected log page into non-volatile storage. The list parameters are formatted in binary.
1	xx	1	x	x	Do not save any of the log parameters into non-volatile storage.

Table A.6 lists all possible parameter values that may be controlled by a LOG SELECT command.

Table A.6 – LOG SELECT controller parameter values

LOG SELECT CDB Values	Log Page Parameter Control Byte Value		Device server action
PC field	LP bit	LBIN bit	Updated parameter value usage
00	0	x	The parameter values for all the log parameters in the log page(s) sent to the device server are used as threshold values, unless the LP bit is set.
01	0	x	The parameter values for all the log parameters in the log page(s) sent to the device server are used as cumulative values, unless the LP bit is set.
10	0	x	The device server sets the current threshold values to the default threshold values for all the log parameters specified in the log page(s) sent during a LOG SELECT command, unless the LP bit is set.
11	0	x	The device server sets the current cumulative values to the default cumulative values for all the log parameters specified in the log page(s) sent during a LOG SELECT command, unless the LP bit is set.
xx	1	0	The device server replaces the current list parameter with the list parameter sent to the device server. The list parameters are formatted as ASCII graphic codes.
xx	1	1	The device server replaces the current list parameter with the list parameter sent to the device server. The list parameters are formatted in binary.

A.4 Exception conditions during logging

The logging operations may be setup to keep track of many different vendor-specific items. This clause describes how a device server informs an application client when a log reaches a critical point, thereby creating an exception condition.

Tables A.7 and A.8 list the definitions of the parameter control byte of the log parameter. Table A.7 lists parameter control byte values that affect parameter saving. Table A.8 lists parameter control byte values that affect parameter updating and reporting.

Table A.7 – Log Parameter Control Byte saving definitions

Parameter Control Byte values		**Control Mode Page (0Ah)**	
DS bit	TSD bit	GLTSD bit	**Description**
0	-	-	Indicates that the device server supports saving for of the log parameter.
1	-	-	Indicates that the device server does not support saving of the log parameter in response to a LOG SELECT or LOG SENSE command.
-	0	0	Indicates that the device server provides a target-defined method of saving log parameters.
-	1	0	Indicates that either the device server does not provide a target-defined method for saving log parameters or the target-defined method has been disabled by an application client.
-	x	1	Indicates that either the device server does not provide a target-defined method for saving log parameters or the target-defined method has been disabled by an application client.

Table A.8 – Log Parameter Control Byte updating definitions

Parameter Control Byte values					
DU bit	ETC bit	TMC field	LP bit	LBIN bit	**Description**
0	-	--	-	-	Indicates that the device server updates the log parameter value to reflect all events that should be noted by that log parameter.
1	-	--	-	-	Indicates that the device server does not update the log parameter value except in response to a LOG SELECT command that specifies a new value the log parameter.
-	0	--	-	-	Indicates that a comparison between the threshold value and the cumulative value is not performed.
-	1	--	-	-	Indicates that a comparison to the threshold value is performed whenever the cumulative value is updated.
-	-	00	-	-	Indicates that device server informs the application client on every update to the cumulative value.
-	-	01	-	-	Indicates that device server informs the application client on every time the cumulative value is equal to the threshold value.
-	-	10	-	-	Indicates that device server informs the application client on every time the cumulative value is not equal to the threshold value.
-	-	11	-	-	Indicates that device server informs the application client on every time the cumulative value is greater than the threshold value.
-	-	--	0	x	Indicates that the log parameter is a data counter.
-	-	--	1	0	Indicates that the log parameter is a list parameter and the list parameter is formatted as ASCII graphic codes.
-	-	--	1	1	Indicates that the log parameter is a list parameter and the list parameter is formatted in binary.

Table A.9 describes the device server actions associated with logging exception conditions.

Table A.9 – Logging exception conditions

Log Page Parameter Control Byte values				**Control Mode Page (0Ah)**	**Device server action**
DU bit	ETC bit	TMC field	LP bit	RLEC bit	Exception condition actions
x	x	xx	x	0	No logging activities will cause an ACA condition or a Unit Attention condition.
x	0	GT 0	1	x	This is an illegal condition.
x	1	xx	1	x	This is an illegal condition.
0	1	xx	0	1	Follow pseudocode 1 (see A.4.1)
0	0	NV	0	1	Follow pseudocode 2 (see A.4.2)
0	0	00	1	1	Follow pseudocode 3 (see A.4.3)

The pseudocode in A.4.1 through A.4.3 assumes that ACA is implemented and requested in the CDB control byte. If this is not the case, the implementation may be based on the SCSI-2 TIB**[1]** or other applicable standards.

A.4.1 Pseudocode 1

IF the threshold condition as defined by the TMC field is met:

a) IF there is an active task
   a) Complete the active task
   b) If an ACA condition exists wait for it to be cleared
   END
b) Issue a Unit Attention condition to all initiators that have set the RLEC bit to one
c) IF the Unit Attention condition is ignored
   a) Continue normal operations until the threshold condition is met again

END

[1] TIB for IT - Procedures for Logging Operations (X3-131 - 1994 / TIB-1).

A.4.2 Pseudocode 2

IF a log counter reaches its maximum value:
- a) Set DU to 1
- b) IF there is no active task
  - a) Wait until there is an active task

  END
- c) Complete the active task
- d) IF no ACA condition exists
  - a) Create an ACA condition with a sense key of RECOVERED ERROR and additional sense data of LOG EXCEPTION, COUNT AT MAXIMUM

  END
- e) Wait for the ACA condition to be cleared
- f) IF the cause of the counter reaching maximum is not cleared by the application client
  - a) Do not create an ACA condition and do not increment the counter

  END

END

A.4.3 Pseudocode 3

IF the log of parameters if full:
- a) Place the new log parameter code value into the lowest parameter code value position (wrap-around the parameter codes)
- b) IF there is no active task
  - a) Wait until there is an active task

  END
- c) Complete the active task
- d) IF no ACA condition exists
  - a) Create an ACA condition with a sense key of RECOVERED ERROR and additional sense data of LOG EXCEPTION, LIST CODES EXHAUSTED

  END
- e) Wait for the ACA condition to be cleared
- f) IF the cause of the log of parameters filling is not cleared by the application client
  - a) Create an ACA condition every time an entry is placed into the log of parameters

  END

END

Annex B
(informative)

Numeric order codes

This annex contains SCSI-3 additional sense codes, operation codes, log page codes, and mode page codes in numeric order as a reference. In the event of a conflict with between the codes in this annex and the codes in the body of this standard, the codes in the body are correct.

The information in this annex was complete and accurate at the time of publication. However, the information is subject to change. Technical Committee T10 of Accredited Standards Committee NCITS maintains an electronic copy of this information on its world wide web site (http://www.symbios.com/x3t10). In the event that the T10 world wide web site is no longer active, access may be possible via the NCITS world wide web site (http://www.x3.org).

Table B.1 is a numerical order listing of the additional sense codes and the additional sense code qualifiers.

Table B.1 – ASC and ASCQ assignments

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)              blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ DTLPWRSOMCAE  Description
--- ---- ------------  -----------
 00  00  DTLPWRSOMCAE  NO ADDITIONAL SENSE INFORMATION
 00  01   T            FILEMARK DETECTED
 00  02   T    S       END-OF-PARTITION/MEDIUM DETECTED
 00  03   T            SETMARK DETECTED
 00  04   T    S       BEGINNING-OF-PARTITION/MEDIUM DETECTED
 00  05   T    S       END-OF-DATA DETECTED
 00  06  DTLPWRSOMCAE  I/O PROCESS TERMINATED
 00  11       R        AUDIO PLAY OPERATION IN PROGRESS
 00  12       R        AUDIO PLAY OPERATION PAUSED
 00  13       R        AUDIO PLAY OPERATION SUCCESSFULLY COMPLETED
 00  14       R        AUDIO PLAY OPERATION STOPPED DUE TO ERROR
 00  15       R        NO CURRENT AUDIO STATUS TO RETURN
 00  16  DTLPWRSOMCAE  OPERATION IN PROGRESS
 00  17  DTL WRSOM AE  CLEANING REQUESTED
 01  00  D   W  O      NO INDEX/SECTOR SIGNAL
 02  00  D   WR OM     NO SEEK COMPLETE
 03  00  DTL W SO      PERIPHERAL DEVICE WRITE FAULT
 03  01   T            NO WRITE CURRENT
 03  02   T            EXCESSIVE WRITE ERRORS
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)              blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ  DTLPWRSOMCAE  Description
--- ---- ------------ ------------------------------------------------
 04  00  DTLPWRSOMCAE  LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
 04  01  DTLPWRSOMCAE  LOGICAL UNIT IS IN PROCESS OF BECOMING READY
 04  02  DTLPWRSOMCAE  LOGICAL UNIT NOT READY, INITIALIZING CMD. REQUIRED
 04  03  DTLPWRSOMCAE  LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
 04  04  DTL    O      LOGICAL UNIT NOT READY, FORMAT IN PROGRESS
 04  05            A   LOGICAL UNIT NOT READY, REBUILD IN PROGRESS
 04  06            A   LOGICAL UNIT NOT READY, RECALCULATION IN PROGRESS
 04  07  DTLPWRSOMCAE  LOGICAL UNIT NOT READY, OPERATION IN PROGRESS
 04  08       R        LOGICAL UNIT NOT READY, LONG WRITE IN PROGRESS
 05  00  DTL WRSOMCAE  LOGICAL UNIT DOES NOT RESPOND TO SELECTION
 06  00  D   WR OM     NO REFERENCE POSITION FOUND
 07  00  DTL WRSOM     MULTIPLE PERIPHERAL DEVICES SELECTED
 08  00  DTL WRSOMCAE  LOGICAL UNIT COMMUNICATION FAILURE
 08  01  DTL WRSOMCAE  LOGICAL UNIT COMMUNICATION TIME-OUT
 08  02  DTL WRSOMCAE  LOGICAL UNIT COMMUNICATION PARITY ERROR
 08  03  DT  R OM      LOGICAL UNIT COMMUNICATION CRC ERROR (ULTRA-DMA/32)
 09  00  DT  WR O      TRACK FOLLOWING ERROR
 09  01      WR O      TRACKING SERVO FAILURE
 09  02      WR O      FOCUS SERVO FAILURE
 09  03      WR O      SPINDLE SERVO FAILURE
 09  04  DT  WR O      HEAD SELECT FAULT
 0A  00  DTLPWRSOMCAE  ERROR LOG OVERFLOW
 0B  00  DTLPWRSOMCAE  WARNING
 0B  01  DTLPWRSOMCAE  WARNING - SPECIFIED TEMPERATURE EXCEEDED
 0B  02  DTLPWRSOMCAE  WARNING - ENCLOSURE DEGRADED
 0C  00   T   RS       WRITE ERROR
 0C  01                WRITE ERROR - RECOVERED WITH AUTO REALLOCATION
 0C  02  D   W  O      WRITE ERROR - AUTO REALLOCATION FAILED
 0C  03  D   W  O      WRITE ERROR - RECOMMEND REASSIGNMENT
 0C  04  DT  W  O      COMPRESSION CHECK MISCOMPARE ERROR
 0C  05  DT  W  O      DATA EXPANSION OCCURRED DURING COMPRESSION
 0C  06  DT  W  O      BLOCK NOT COMPRESSIBLE
 0C  07       R        WRITE ERROR - RECOVERY NEEDED
 0C  08       R        WRITE ERROR - RECOVERY FAILED
 0C  09       R        WRITE ERROR - LOSS OF STREAMING
 0C  0A       R        WRITE ERROR - PADDING BLOCKS ADDED
 0D  00
 0E  00
 0F  00
 10  00  D   W  O      ID CRC OR ECC ERROR
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)             blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ  DTLPWRSOMCAE  Description
--- ---- ------------  -----------
 11  00  DT  WRSO      UNRECOVERED READ ERROR
 11  01  DT  W SO      READ RETRIES EXHAUSTED
 11  02  DT  W SO      ERROR TOO LONG TO CORRECT
 11  03  DT  W SO      MULTIPLE READ ERRORS
 11  04  D   W  O      UNRECOVERED READ ERROR - AUTO REALLOCATE FAILED
 11  05      WR O      L-EC UNCORRECTABLE ERROR
 11  06      WR O      CIRC UNRECOVERED ERROR
 11  07      W  O      DATA RE-SYNCHRONIZATION ERROR
 11  08   T            INCOMPLETE BLOCK READ
 11  09   T            NO GAP FOUND
 11  0A  DT     O      MISCORRECTED ERROR
 11  0B  D   W  O      UNRECOVERED READ ERROR - RECOMMEND REASSIGNMENT
 11  0C  D   W  O      UNRECOVERED READ ERROR - RECOMMEND REWRITE THE DATA
 11  0D  DT  WR O      DE-COMPRESSION CRC ERROR
 11  0E  DT  WR O      CANNOT DECOMPRESS USING DECLARED ALGORITHM
 11  0F       R        ERROR READING UPC/EAN NUMBER
 11  10       R        ERROR READING ISRC NUMBER
 11  11       R        READ ERROR - LOSS OF STREAMING
 12  00  D   W  O      ADDRESS MARK NOT FOUND FOR ID FIELD
 13  00  D   W  O      ADDRESS MARK NOT FOUND FOR DATA FIELD
 14  00  DTL WRSO      RECORDED ENTITY NOT FOUND
 14  01  DT  WR O      RECORD NOT FOUND
 14  02   T            FILEMARK OR SETMARK NOT FOUND
 14  03   T            END-OF-DATA NOT FOUND
 14  04   T            BLOCK SEQUENCE ERROR
 14  05  DT  W  O      RECORD NOT FOUND - RECOMMEND REASSIGNMENT
 14  06  DT  W  O      RECORD NOT FOUND - DATA AUTO-REALLOCATED
 15  00  DTL WRSOM     RANDOM POSITIONING ERROR
 15  01  DTL WRSOM     MECHANICAL POSITIONING ERROR
 15  02  DT  WR O      POSITIONING ERROR DETECTED BY READ OF MEDIUM
 16  00  D   W  O      DATA SYNCHRONIZATION MARK ERROR
 16  01  D   W  O      DATA SYNC ERROR - DATA REWRITTEN
 16  02  D   W  O      DATA SYNC ERROR - RECOMMEND REWRITE
 16  03  D   W  O      DATA SYNC ERROR - DATA AUTO-REALLOCATED
 16  04  D   W  O      DATA SYNC ERROR - RECOMMEND REASSIGNMENT
 17  00  DT  WRSO      RECOVERED DATA WITH NO ERROR CORRECTION APPLIED
 17  01  DT  WRSO      RECOVERED DATA WITH RETRIES
 17  02  DT  WR O      RECOVERED DATA WITH POSITIVE HEAD OFFSET
 17  03  DT  WR O      RECOVERED DATA WITH NEGATIVE HEAD OFFSET
 17  04      WR O      RECOVERED DATA WITH RETRIES AND/OR CIRC APPLIED
 17  05  D   WR O      RECOVERED DATA USING PREVIOUS SECTOR ID
 17  06  D   W  O      RECOVERED DATA WITHOUT ECC - DATA AUTO-REALLOCATED
 17  07  D   W  O      RECOVERED DATA WITHOUT ECC - RECOMMEND REASSIGNMENT
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)              blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ  DTLPWRSOMCAE  Description
--- ---- ------------ -----------
 17  08  D   W  O     RECOVERED DATA WITHOUT ECC - RECOMMEND REWRITE
 17  09  D   W  O     RECOVERED DATA WITHOUT ECC - DATA REWRITTEN
 18  00  DT  WR O     RECOVERED DATA WITH ERROR CORRECTION APPLIED
 18  01  D   WR O     RECOVERED DATA WITH ERROR CORR. & RETRIES APPLIED
 18  02  D   WR O     RECOVERED DATA - DATA AUTO-REALLOCATED
 18  03       R       RECOVERED DATA WITH CIRC
 18  04       R       RECOVERED DATA WITH L-EC
 18  05  D   WR O     RECOVERED DATA - RECOMMEND REASSIGNMENT
 18  06  D   WR O     RECOVERED DATA - RECOMMEND REWRITE
 18  07  D   W  O     RECOVERED DATA WITH ECC - DATA REWRITTEN
 19  00  D      O     DEFECT LIST ERROR
 19  01  D      O     DEFECT LIST NOT AVAILABLE
 19  02  D      O     DEFECT LIST ERROR IN PRIMARY LIST
 19  03  D      O     DEFECT LIST ERROR IN GROWN LIST
 1A  00  DTLPWRSOMCAE  PARAMETER LIST LENGTH ERROR
 1B  00  DTLPWRSOMCAE  SYNCHRONOUS DATA TRANSFER ERROR
 1C  00  D      O     DEFECT LIST NOT FOUND
 1C  01  D      O     PRIMARY DEFECT LIST NOT FOUND
 1C  02  D      O     GROWN DEFECT LIST NOT FOUND
 1D  00  D   W  O     MISCOMPARE DURING VERIFY OPERATION
 1E  00  D   W  O     RECOVERED ID WITH ECC CORRECTION
 1F  00  D      O     PARTIAL DEFECT LIST TRANSFER
 20  00  DTLPWRSOMCAE  INVALID COMMAND OPERATION CODE
 21  00  DT  WR OM    LOGICAL BLOCK ADDRESS OUT OF RANGE
 21  01  DT  WR OM    INVALID ELEMENT ADDRESS
 22  00  D             ILLEGAL FUNCTION (USE 20 00, 24 00, OR 26 00)
 23  00
 24  00  DTLPWRSOMCAE  INVALID FIELD IN CDB
 25  00  DTLPWRSOMCAE  LOGICAL UNIT NOT SUPPORTED
 26  00  DTLPWRSOMCAE  INVALID FIELD IN PARAMETER LIST
 26  01  DTLPWRSOMCAE  PARAMETER NOT SUPPORTED
 26  02  DTLPWRSOMCAE  PARAMETER VALUE INVALID
 26  03  DTLPWRSOMCAE  THRESHOLD PARAMETERS NOT SUPPORTED
 26  04  DTLPWRSOMCAE  INVALID RELEASE OF ACTIVE PERSISTENT RESERVATION
 27  00  DT  W  O     WRITE PROTECTED
 27  01  DT  W  O     HARDWARE WRITE PROTECTED
 27  02  DT  W  O     LOGICAL UNIT SOFTWARE WRITE PROTECTED
 27  03   T            ASSOCIATED WRITE PROTECT
 27  04   T            PERSISTENT WRITE PROTECT
 27  05   T            PERMANENT WRITE PROTECT
 28  00  DTLPWRSOMCAE  NOT READY TO READY CHANGE, MEDIUM MAY HAVE CHANGED
 28  01  DT  WR OM    IMPORT OR EXPORT ELEMENT ACCESSED
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)             blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ DTLPWRSOMCAE  Description
 29  00  DTLPWRSOMCAE  POWER ON, RESET, OR BUS DEVICE RESET OCCURRED
 29  01  DTLPWRSOMCAE  POWER ON OCCURRED
 29  02  DTLPWRSOMCAE  SCSI BUS RESET OCCURRED
 29  03  DTLPWRSOMCAE  BUS DEVICE RESET FUNCTION OCCURRED
 29  04  DTLPWRSOMCAE  DEVICE INTERNAL RESET
 2A  00  DTL WRSOMCAE  PARAMETERS CHANGED
 2A  01  DTL WRSOMCAE  MODE PARAMETERS CHANGED
 2A  02  DTL WRSOMCAE  LOG PARAMETERS CHANGED
 2A  03  DTLPWRSOMCAE  RESERVATIONS PREEMPTED
 2B  00  DTLPWRSO C    COPY CANNOT EXECUTE SINCE HOST CANNOT DISCONNECT
 2C  00  DTLPWRSOMCAE  COMMAND SEQUENCE ERROR
 2C  01        S       TOO MANY WINDOWS SPECIFIED
 2C  02        S       INVALID COMBINATION OF WINDOWS SPECIFIED
 2C  03       R        CURRENT PROGRAM AREA IS NOT EMPTY
 2C  04       R        CURRENT PROGRAM AREA IS EMPTY
 2D  00   T            OVERWRITE ERROR ON UPDATE IN PLACE
 2E  00
 2F  00  DTLPWRSOMCAE  COMMANDS CLEARED BY ANOTHER INITIATOR
 30  00  DT  WR OM     INCOMPATIBLE MEDIUM INSTALLED
 30  01  DT  WR O      CANNOT READ MEDIUM - UNKNOWN FORMAT
 30  02  DT  WR O      CANNOT READ MEDIUM - INCOMPATIBLE FORMAT
 30  03  DT            CLEANING CARTRIDGE INSTALLED
 30  04  DT  WR O      CANNOT WRITE MEDIUM - UNKNOWN FORMAT
 30  05  DT  WR O      CANNOT WRITE MEDIUM - INCOMPATIBLE FORMAT
 30  06  DT  W  O      CANNOT FORMAT MEDIUM - INCOMPATIBLE MEDIUM
 30  07  DTL WRSOM AE  CLEANING FAILURE
 30  08       R        CANNOT WRITE - APPLICATION CODE MISMATCH
 30  09       R        CURRENT SESSION NOT FIXATED FOR APPEND
 31  00  DT  WR O      MEDIUM FORMAT CORRUPTED
 31  01  D L  R O      FORMAT COMMAND FAILED
 32  00  D   W  O      NO DEFECT SPARE LOCATION AVAILABLE
 32  01  D   W  O      DEFECT LIST UPDATE FAILURE
 33  00   T            TAPE LENGTH ERROR
 34  00  DTLPWRSOMCAE  ENCLOSURE FAILURE
 35  00  DTLPWRSOMCAE  ENCLOSURE SERVICES FAILURE
 35  01  DTLPWRSOMCAE  UNSUPPORTED ENCLOSURE FUNCTION
 35  02  DTLPWRSOMCAE  ENCLOSURE SERVICES UNAVAILABLE
 35  03  DTLPWRSOMCAE  ENCLOSURE SERVICES TRANSFER FAILURE
 35  04  DTLPWRSOMCAE  ENCLOSURE SERVICES TRANSFER REFUSED
 36  00    L           RIBBON, INK, OR TONER FAILURE
 37  00  DTL WRSOMCAE  ROUNDED PARAMETER
 38  00
 39  00  DTL WRSOMCAE  SAVING PARAMETERS NOT SUPPORTED
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)              blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ DTLPWRSOMCAE  Description
--- ---- ----------- ---------------------------------------------------
 3A  00  DTL WRSOM    MEDIUM NOT PRESENT
 3A  01  DT  WR OM    MEDIUM NOT PRESENT - TRAY CLOSED
 3A  02  DT  WR OM    MEDIUM NOT PRESENT - TRAY OPEN
 3B  00   TL          SEQUENTIAL POSITIONING ERROR
 3B  01   T           TAPE POSITION ERROR AT BEGINNING-OF-MEDIUM
 3B  02   T           TAPE POSITION ERROR AT END-OF-MEDIUM
 3B  03    L          TAPE OR ELECTRONIC VERTICAL FORMS UNIT NOT READY
 3B  04    L          SLEW FAILURE
 3B  05    L          PAPER JAM
 3B  06    L          FAILED TO SENSE TOP-OF-FORM
 3B  07    L          FAILED TO SENSE BOTTOM-OF-FORM
 3B  08   T           REPOSITION ERROR
 3B  09        S       READ PAST END OF MEDIUM
 3B  0A        S       READ PAST BEGINNING OF MEDIUM
 3B  0B        S       POSITION PAST END OF MEDIUM
 3B  0C   T    S       POSITION PAST BEGINNING OF MEDIUM
 3B  0D  DT  WR OM     MEDIUM DESTINATION ELEMENT FULL
 3B  0E  DT  WR OM     MEDIUM SOURCE ELEMENT EMPTY
 3B  0F       R        END OF MEDIUM REACHED
 3B  11  DT  WR OM     MEDIUM MAGAZINE NOT ACCESSIBLE
 3B  12  DT  WR OM     MEDIUM MAGAZINE REMOVED
 3B  13  DT  WR OM     MEDIUM MAGAZINE INSERTED
 3B  14  DT  WR OM     MEDIUM MAGAZINE LOCKED
 3B  15  DT  WR OM     MEDIUM MAGAZINE UNLOCKED
 3C  00
 3D  00  DTLPWRSOMCAE  INVALID BITS IN IDENTIFY MESSAGE
 3E  00  DTLPWRSOMCAE  LOGICAL UNIT HAS NOT SELF-CONFIGURED YET
 3E  01            A   LOGICAL UNIT FAILURE
 3E  02            A   TIMEOUT ON LOGICAL UNIT
 3F  00  DTLPWRSOMCAE  TARGET OPERATING CONDITIONS HAVE CHANGED
 3F  01  DTLPWRSOMCAE  MICROCODE HAS BEEN CHANGED
 3F  02  DTLPWRSOMC    CHANGED OPERATING DEFINITION
 3F  03  DTLPWRSOMCAE  INQUIRY DATA HAS CHANGED
 40  00  D             RAM FAILURE (SHOULD USE 40 NN)
 40  NN  DTLPWRSOMCAE  DIAGNOSTIC FAILURE ON COMPONENT NN (80H-FFH)
 41  00  D             DATA PATH FAILURE (SHOULD USE 40 NN)
 42  00  D             POWER-ON OR SELF-TEST FAILURE (SHOULD USE 40 NN)
 43  00  DTLPWRSOMCAE  MESSAGE ERROR
 44  00  DTLPWRSOMCAE  INTERNAL TARGET FAILURE
 45  00  DTLPWRSOMCAE  SELECT OR RESELECT FAILURE
 46  00  DTLPWRSOMC    UNSUCCESSFUL SOFT RESET
 47  00  DTLPWRSOMCAE  SCSI PARITY ERROR
 48  00  DTLPWRSOMCAE  INITIATOR DETECTED ERROR MESSAGE RECEIVED
 49  00  DTLPWRSOMCAE  INVALID MESSAGE ERROR
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)             blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ  DTLPWRSOMCAE  Description
---------------------------------------------------------------------------
 4A  00   DTLPWRSOMCAE  COMMAND PHASE ERROR
 4B  00   DTLPWRSOMCAE  DATA PHASE ERROR
 4C  00   DTLPWRSOMCAE  LOGICAL UNIT FAILED SELF-CONFIGURATION
 4D  NN   DTLPWRSOMCAE  TAGGED OVERLAPPED COMMANDS (NN = QUEUE TAG)
 4E  00   DTLPWRSOMCAE  OVERLAPPED COMMANDS ATTEMPTED
 4F  00
 50  00    T            WRITE APPEND ERROR
 50  01    T            WRITE APPEND POSITION ERROR
 50  02    T            POSITION ERROR RELATED TO TIMING
 51  00    T     O      ERASE FAILURE
 52  00    T            CARTRIDGE FAULT
 53  00   DTL WRSOM     MEDIA LOAD OR EJECT FAILED
 53  01    T            UNLOAD TAPE FAILURE
 53  02   DT  WR OM     MEDIUM REMOVAL PREVENTED
 54  00      P          SCSI TO HOST SYSTEM INTERFACE FAILURE
 55  00      P          SYSTEM RESOURCE FAILURE
 55  01   D      O      SYSTEM BUFFER FULL
 56  00
 57  00       R         UNABLE TO RECOVER TABLE-OF-CONTENTS
 58  00         O       GENERATION DOES NOT EXIST
 59  00         O       UPDATED BLOCK READ
 5A  00   DTLPWRSOM     OPERATOR REQUEST OR STATE CHANGE INPUT
 5A  01   DT  WR OM     OPERATOR MEDIUM REMOVAL REQUEST
 5A  02   DT  W  O      OPERATOR SELECTED WRITE PROTECT
 5A  03   DT  W  O      OPERATOR SELECTED WRITE PERMIT
 5B  00   DTLPWRSOM     LOG EXCEPTION
 5B  01   DTLPWRSOM     THRESHOLD CONDITION MET
 5B  02   DTLPWRSOM     LOG COUNTER AT MAXIMUM
 5B  03   DTLPWRSOM     LOG LIST CODES EXHAUSTED
 5C  00   D      O      RPL STATUS CHANGE
 5C  01   D      O      SPINDLES SYNCHRONIZED
 5C  02   D      O      SPINDLES NOT SYNCHRONIZED
 5D  00   DTLPWRSOMCAE  FAILURE PREDICTION THRESHOLD EXCEEDED
 5D  FF   DTLPWRSOMCAE  FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE)
 5E  00   DTLPWRSO CA   LOW POWER CONDITION ON
 5E  01   DTLPWRSO CA   IDLE CONDITION ACTIVATED BY TIMER
 5E  02   DTLPWRSO CA   STANDBY CONDITION ACTIVATED BY TIMER
 5E  03   DTLPWRSO CA   IDLE CONDITION ACTIVATED BY COMMAND
 5E  04   DTLPWRSO CA   STANDBY CONDITION ACTIVATED BY COMMAND
 5F  00
 60  00         S       LAMP FAILURE
 61  00         S       VIDEO ACQUISITION ERROR
 61  01         S       UNABLE TO ACQUIRE VIDEO
 61  02         S       OUT OF FOCUS
```

(continued)

Table B.1 – ASC and ASCQ assignments (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)              blank = reserved
          . L - PRINTER DEVICE (SSC)                    not blank = allowed
          .  P - PROCESSOR DEVICE (SPC)
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
          .  . R - CD DEVICE (MMC)
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
ASC ASCQ DTLPWRSOMCAE  Description
--- ---- ----------- ----------------------------------------------------
 62  00        S      SCAN HEAD POSITIONING ERROR
 63  00       R       END OF USER AREA ENCOUNTERED ON THIS TRACK
 63  01       R       PACKET DOES NOT FIT IN AVAILABLE SPACE
 64  00       R       ILLEGAL MODE FOR THIS TRACK
 64  01       R       INVALID PACKET SIZE
 65  00  DTLPWRSOMCAE  VOLTAGE FAULT
 66  00        S      AUTOMATIC DOCUMENT FEEDER COVER UP
 66  01        S      AUTOMATIC DOCUMENT FEEDER LIFT UP
 66  02        S      DOCUMENT JAM IN AUTOMATIC DOCUMENT FEEDER
 66  03        S      DOCUMENT MISS FEED AUTOMATIC IN DOCUMENT FEEDER
 67  00            A  CONFIGURATION FAILURE
 67  01            A  CONFIGURATION OF INCAPABLE LOGICAL UNITS FAILED
 67  02            A  ADD LOGICAL UNIT FAILED
 67  03            A  MODIFICATION OF LOGICAL UNIT FAILED
 67  04            A  EXCHANGE OF LOGICAL UNIT FAILED
 67  05            A  REMOVE OF LOGICAL UNIT FAILED
 67  06            A  ATTACHMENT OF LOGICAL UNIT FAILED
 67  07            A  CREATION OF LOGICAL UNIT FAILED
 68  00            A  LOGICAL UNIT NOT CONFIGURED
 69  00            A  DATA LOSS ON LOGICAL UNIT
 69  01            A  MULTIPLE LOGICAL UNIT FAILURES
 69  02            A  PARITY/DATA MISMATCH
 6A  00            A  INFORMATIONAL, REFER TO LOG
 6B  00            A  STATE CHANGE HAS OCCURRED
 6B  01            A  REDUNDANCY LEVEL GOT BETTER
 6B  02            A  REDUNDANCY LEVEL GOT WORSE
 6C  00            A  REBUILD FAILURE OCCURRED
 6D  00            A  RECALCULATE FAILURE OCCURRED
 6E  00            A  COMMAND TO LOGICAL UNIT FAILED
 6F  00
 70  NN   T            DECOMPRESSION EXCEPTION SHORT ALGORITHM ID OF NN
 71  00   T            DECOMPRESSION EXCEPTION LONG ALGORITHM ID
 72  00       R       SESSION FIXATION ERROR
 72  01       R       SESSION FIXATION ERROR WRITING LEAD-IN
 72  02       R       SESSION FIXATION ERROR WRITING LEAD-OUT
 72  03       R       SESSION FIXATION ERROR - INCOMPLETE TRACK IN SESSION
 72  04       R       EMPTY OR PARTIALLY WRITTEN RESERVED TRACK
 73  00       R       CD CONTROL ERROR
 73  01       R       POWER CALIBRATION AREA ALMOST FULL
 73  02       R       POWER CALIBRATION AREA IS FULL
 73  03       R       POWER CALIBRATION AREA ERROR
 73  04       R       PROGRAM MEMORY AREA UPDATE FAILURE
 73  05       R       PROGRAM MEMORY AREA IS FULL
```

(continued)

Table B.1 – ASC and ASCQ assignments (concluded)

ASC	ASCQ	DTLPWRSOMCAE	Description
		D - DIRECT ACCESS DEVICE (SBC)	Device column key
		.T - SEQUENTIAL ACCESS DEVICE (SSC)	blank = reserved
		. L - PRINTER DEVICE (SSC)	not blank = allowed
		. P - PROCESSOR DEVICE (SPC)	
		. .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)	
		. . R - CD DEVICE (MMC)	
		. . S - SCANNER DEVICE (SGC)	
		. . .O - OPTICAL MEMORY DEVICE (SBC)	
		. . . M - MEDIA CHANGER DEVICE (SMC)	
		. . . C - COMMUNICATION DEVICE (SSC)	
		. . . .A - STORAGE ARRAY DEVICE (SCC)	
		. . . . E - ENCLOSURE SERVICES DEVICE (SES)	
		. . . .	
74	00		
75	00		
76	00		
77	00		
78	00		
79	00		
7A	00		
7B	00		
7C	00		
7D	00		
7E	00		
7F	00		
80	*xx*		
THROUGH			Vendor-specific.
FF	*xx*		
xx	80		
THROUGH			Vendor-specific QUALIFICATION OF STANDARD ASC.
xx	FF		
			ALL CODES NOT SHOWN ARE RESERVED.

Table B.2 is a numerical order listing of the command operation codes.

Table B.2 – SCSI-3 Operation Codes

```
          D - DIRECT ACCESS DEVICE (SBC)                    Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)               M = Mandatory
          . L - PRINTER DEVICE (SSC)                        O = Optional
          .  P - PROCESSOR DEVICE (SPC)                     V = Vendor-specific
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)     R = Reserved
          .  . R - CD DEVICE (MMC)                          Z = Obsolete
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
       OP DTLPWRSOMCAE Description
       ---|----------|-----------------------------
       00 MMMMMMMMMMM  TEST UNIT READY
       01  M           REWIND
       01 Z V ZO ZO    REZERO UNIT
       02 VVVVVV  V
       03 MMMMMMMMMMM  REQUEST SENSE
       04 M    O O     FORMAT UNIT
       04  O           FORMAT MEDIUM
       04   O          FORMAT
       05 VMVVVV  V    READ BLOCK LIMITS
       06 VVVVVV  V
       07 OVV O  OV    REASSIGN BLOCKS
       07         O    INITIALIZE ELEMENT STATUS
       08 OMV OO OV    READ(06)
       08    O         RECEIVE
       08          M   GET MESSAGE(06)
       09 VVVVVV  V
       0A OM  O  OV    WRITE(06)
       0A    M         SEND(06)
       0A          M   SEND MESSAGE(06)
       0A   M          PRINT
       0B Z   ZO ZV    SEEK(06)
       0B   O          SLEW AND PRINT
       0C VVVVVV  V
       0D VVVVVV  V
       0E VVVVVV  V
       0F VOVVVV  V    READ REVERSE
       10 VM VVV       WRITE FILEMARKS
       10   O O        SYNCHRONIZE BUFFER
       11 VMVVVV       SPACE
       12 MMMMMMMMMMM  INQUIRY
       13 VOVVVV       VERIFY(06)
       14 VOOVVV       RECOVER BUFFERED DATA
       15 OMO OOOOOOOO MODE SELECT(06)
       16 MMMOMMMM   O RESERVE(06)
       16         M    RESERVE ELEMENT(06)
       17 MMMOMMMM   O RELEASE(06)
       17         M    RELEASE ELEMENT(06)
       18 OOOOOOOO     COPY
       19 VMVVVV       ERASE
       1A OMO OOOOOOOO MODE SENSE(06)
```

(continued)

Table B.2 – SCSI-3 Operation Codes (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                    Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)               M = Mandatory
          . L - PRINTER DEVICE (SSC)                        O = Optional
          .  P - PROCESSOR DEVICE (SPC)                     V = Vendor-specific
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)     R = Reserved
          .  . R - CD DEVICE (MMC)                          Z = Obsolete
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
        OP DTLPWRSOMCAE Description
        ---------------------------------------------------------
        1B O   OM O     STOP START UNIT
        1B  O           LOAD UNLOAD
        1B       O      SCAN
        1B   O          STOP PRINT
        1C OOOOOOOOOO M RECEIVE DIAGNOSTIC RESULTS
        1D MMMMMMMMMMM SEND DIAGNOSTIC
        1E OO  OM OO    PREVENT ALLOW MEDIUM REMOVAL
        1F
        20 V   VV V
        21 V   VV V
        22 V   VV V
        23 V   VV V
        24 V   VVM      SET WINDOW
        25 M   M  M     READ CAPACITY
        25      M       READ CD RECORDED CAPACITY
        25       O      GET WINDOW
        26 V   VV
        27 V   VV
        28 M   MMMM     READ(10)
        28           O  GET MESSAGE(10)
        29 V   VV O     READ GENERATION
        2A M   MM M     WRITE(10)
        2A       O      SEND(10)
        2A          O   SEND MESSAGE(10)
        2B O   OM O     SEEK(10)
        2B  O           LOCATE
        2B         O    POSITION TO ELEMENT
        2C V       O    ERASE(10)
        2D V   O  O     READ UPDATED BLOCK
        2E O   O  O     WRITE AND VERIFY(10)
        2F O   OO O     VERIFY(10)
        30 Z   ZO Z     SEARCH DATA HIGH(10)
        31 Z   ZO Z     SEARCH DATA EQUAL(10)
        31       O      OBJECT POSITION
        32 Z   ZO Z     SEARCH DATA LOW(10)
        33 O   OO O     SET LIMITS(10)
        34 O   OO O     PRE-FETCH
        34  O           READ POSITION
        34       O      GET DATA BUFFER STATUS
        35 O   OM O     SYNCHRONIZE CACHE
        36 O   OO O     LOCK UNLOCK CACHE
        37 O       O    READ DEFECT DATA(10)
        38     O  O     MEDIUM SCAN
```

(continued)

Table B.2 – SCSI-3 Operation Codes (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                    Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)               M = Mandatory
          . L - PRINTER DEVICE (SSC)                        O = Optional
          .  P - PROCESSOR DEVICE (SPC)                     V = Vendor-specific
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)     R = Reserved
          .  . R - CD DEVICE (MMC)                          Z = Obsolete
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
       OP DTLPWRSOMCAE Description
       ---|----------|-----------------------------------------
       39 OOOOOOOO     COMPARE
       3A OOOOOOOO     COPY AND VERIFY
       3B OOOOOOOOOO O WRITE BUFFER
       3C OOOOOOOOOO   READ BUFFER
       3D     O  O     UPDATE BLOCK
       3E O   OO O     READ LONG
       3F O   O  O     WRITE LONG
       40 OOOOOOOOOO   CHANGE DEFINITION
       41 O            WRITE SAME
       42      M       READ SUB-CHANNEL
       43      M       READ TOC/PMA/ATIP {MMC Proposed}
       44  M           REPORT DENSITY SUPPORT
       44      M       READ HEADER
       45      O       PLAY AUDIO(10)
       46
       47      O       PLAY AUDIO MSF
       48      O       PLAY AUDIO TRACK INDEX
       49      O       PLAY TRACK RELATIVE(10)
       4A
       4B      O       PAUSE/RESUME
       4C OOOOOOOOOO   LOG SELECT
       4D OOOOOOOOOO   LOG SENSE
       4E      O       STOP PLAY/SCAN {MMC Proposed}
       4F
       50 O            XDWRITE(10)
       51 O            XPWRITE(10)
       51      M       READ DISC INFORMATION {MMC Proposed}
       52 O            XDREAD(10)
       52      M       READ TRACK INFORMATION {MMC Proposed}
       53      M       RESERVE TRACK {MMC Proposed}
       54      O       SEND OPC INFORMATION {MMC Proposed}
       55 OOO OOOOOOOO MODE SELECT(10)
       56 MMMOMMMM   O RESERVE(10)
       56         M    RESERVE ELEMENT(10)
       57 MMMOMMMM   O RELEASE(10)
       57         M    RELEASE ELEMENT(10)
       58      O       REPAIR TRACK {MMC Proposed}
       59      O       READ MASTER CUE {MMC Proposed}
       5A OOO OOOOOOOO MODE SENSE(10)
       5B      M       CLOSE TRACK/SESSION {MMC Proposed}
       5C      O       READ BUFFER CAPACITY {MMC Proposed}
       5D      O       SEND CUE SHEET {MMC Proposed}
       5E OOOOOOOOO  O PERSISTENT RESERVE IN
       5F OOOOOOOOO  O PERSISTENT RESERVE OUT
```

(continued)

Table B.2 – SCSI-3 Operation Codes (continued)

```
          D - DIRECT ACCESS DEVICE (SBC)                    Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)               M = Mandatory
          . L - PRINTER DEVICE (SSC)                        O = Optional
          .  P - PROCESSOR DEVICE (SPC)                     V = Vendor-specific
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)     R = Reserved
          .  . R - CD DEVICE (MMC)                          Z = Obsolete
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
       OP DTLPWRSOMCAE Description
       -- ----------- -----------
       80 O           XDWRITE EXTENDED(16)
       81 O           REBUILD(16)
       82 O           REGENERATE(16)
       83
       84
       85
       86
       87
       88
       89
       8A
       8B
       8C
       8D
       8E
       8F
       90
       91
       92
       93
       94
       95
       96
       97
       98
       99
       9A
       9B
       9C
       9D
       9E
       9F
       A0 OOOOOOOOOO  REPORT LUNS
       A1      O      BLANK {MMC Proposed}
       A2      O      WRITE CD MSF {MMC Proposed}
       A3           M MAINTENANCE (IN)
       A4           O MAINTENANCE (OUT)
       A5  O       M  MOVE MEDIUM
       A5      O      PLAY AUDIO(12)
       A6         O   EXCHANGE MEDIUM
       A6      O      LOAD/UNLOAD CD {MMC Proposed}
       A7 OO  OO OO   MOVE MEDIUM ATTACHED
       A8     OM O    READ(12)
```

(continued)

Table B.2 – SCSI-3 Operation Codes (concluded)

```
          D - DIRECT ACCESS DEVICE (SBC)                    Device column key
          .T - SEQUENTIAL ACCESS DEVICE (SSC)               M = Mandatory
          . L - PRINTER DEVICE (SSC)                        O = Optional
          .  P - PROCESSOR DEVICE (SPC)                     V = Vendor-specific
          .  .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)     R = Reserved
          .  . R - CD DEVICE (MMC)                          Z = Obsolete
          .  .  S - SCANNER DEVICE (SGC)
          .  .  .O - OPTICAL MEMORY DEVICE (SBC)
          .  .  . M - MEDIA CHANGER DEVICE (SMC)
          .  .  .  C - COMMUNICATION DEVICE (SSC)
          .  .  .  .A - STORAGE ARRAY DEVICE (SCC)
          .  .  .  . E - ENCLOSURE SERVICES DEVICE (SES)
          .  .  .  .
       OP DTLPWRSOMCAE Description
       -----------------------------------------------------------
       A8           O  GET MESSAGE(12)
       A9       O      PLAY TRACK RELATIVE(12)
       AA      O  O    WRITE(12)
       AA       O      WRITE CD(12) {MMC Proposed}
       AA           O  SEND MESSAGE(12)
       AB
       AC         O    ERASE(12)
       AD
       AE      O  O    WRITE AND VERIFY(12)
       AF      OO O    VERIFY(12)
       B0      ZO Z    SEARCH DATA HIGH(12)
       B1      ZO Z    SEARCH DATA EQUAL(12)
       B2      ZO Z    SEARCH DATA LOW(12)
       B3      OO O    SET LIMITS(12)
       B4 OO   OO OO   READ ELEMENT STATUS ATTACHED
       B5           O  REQUEST VOLUME ELEMENT ADDRESS
       B6           O  SEND VOLUME TAG
       B7         O    READ DEFECT DATA(12)
       B8  O       M   READ ELEMENT STATUS
       B8       O      SET CD SPEED {MMC Proposed}
       B9       M      READ CD MSF {MMC Proposed}
       BA       O      SCAN {MMC Proposed}
       BA            M REDUNDANCY GROUP (IN)
       BB       O      SET CD-ROM SPEED {proposed}
       BB            O REDUNDANCY GROUP (OUT)
       BC       O      PLAY CD {MMC Proposed}
       BC            M SPARE (IN)
       BD       M      MECHANISM STATUS {MMC Proposed}
       BD            O SPARE (OUT)
       BE       O      READ CD {MMC Proposed}
       BE            M VOLUME SET (IN)
       BF            O VOLUME SET (OUT)
```

Table B.3 – SCSI-3 Log Page Codes

Log Page Code	D - DIRECT ACCESS DEVICE (SBC) .T - SEQUENTIAL ACCESS DEVICE (SSC) . L - PRINTER DEVICE (SSC) . P - PROCESSOR DEVICE (SPC) . .W - WRITE ONCE READ MULTIPLE DEVICE (SBC) . . R - CD DEVICE (MMC) . . S - SCANNER DEVICE (SGC) . . .O - OPTICAL MEMORY DEVICE (SBC) . . . M - MEDIA CHANGER DEVICE (SMC) . . . C - COMMUNICATION DEVICE (SSC) . . . .A - STORAGE ARRAY DEVICE (SCC) . . . . E - ENCLOSURE SERVICES DEVICE (SES) . . . . DTLPWRSOMCAE	Description	Device column key blank = reserved not blank = allowed
00h	DTLPWRSOMCAE	Supported log pages	
01h	DTLPWRSO CA	Buffer over-run/under-run page	
02h	DT W O C	Error counter page (write) page	
03h	DT WRSO C	Error counter page (read) page	
04h	T C	Error counter page (read reverse) page	
05h	DT W O C	Error counter page (verify) page	
06h	DTLPWRSOMCAE	Non-medium error page	
07h	DTLPWRSOMCAE	Last n error events page	
08h	DT W O	Format status page	
09h	O	Reserved to the MS59 Std. (contact AIIM C21 comm.)	
0Ah	O	Reserved to the MS59 Std. (contact AIIM C21 comm.)	
0Bh	DTLPWRSOMCAE	Last n deferred error events page	
0Ch	T	Sequential-access Device page	
30h THROUGH 3Eh		Vendor-specific.	
		ALL CODES NOT SHOWN ARE RESERVED.	

Table B.4 – SCSI-3 Mode Page Codes

Mode Page Code	D - DIRECT ACCESS DEVICE (SBC) .T - SEQUENTIAL ACCESS DEVICE (SSC) . L - PRINTER DEVICE (SSC) . P - PROCESSOR DEVICE (SPC) . .W - WRITE ONCE READ MULTIPLE DEVICE (SBC) . . R - CD DEVICE (MMC) . . S - SCANNER DEVICE (SGC) . . .O - OPTICAL MEMORY DEVICE (SBC) . . . M - MEDIA CHANGER DEVICE (SMC) . . . C - COMMUNICATION DEVICE (SSC) . . . .A - STORAGE ARRAY DEVICE (SCC) . . . . E - ENCLOSURE SERVICES DEVICE (SES) DTLPWRSOMCAE	Description
01h	DT WR O	Read-write error recovery mode page
02h	DTL WRSO CAE	Disconnect-reconnect page
03h	D	Format device mode page
03h	L	Parallel printer interface mode page
03h	S	Measurements units mode page
04h	D	Rigid disk geometry mode page
04h	L	Serial printer interface mode page
05h	D	Flexible disk mode page
05h	L	Printer options mode page
06h	W O	Optical memory mode page
07h	D W O	Verify error recover mode page
08h	D WR O	Caching mode page
09h	DTL WRSO CAE	obsolete
0Ah	DTL WRSO CAE	Control mode page
0Bh	D WR O	Medium types supported mode page
0Ch	D	Notch and partition mode page
0Dh	D	Power Condition mode page **[1]**
0Dh	R	CD-ROM mode page
0Eh	R	CD-ROM audio control mode page
0Fh		
10h	D	XOR Control mode page
10h	T	Device configuration mode page
11h	T	Medium partition mode page (1)
12h	T	Medium partition mode page (2)
13h	T	Medium partition mode page (3)
14h	T	Medium partition mode page (4)
15h		
16h		
17h		
18h		
19h		

Device column key
blank = reserved
not blank = allowed

(continued)

Table B.4 – SCSI-3 Mode Page Codes (concluded)

Mode Page Code	DTLPWRSOMCAE	Description
		D - DIRECT ACCESS DEVICE (SBC)
		.T - SEQUENTIAL ACCESS DEVICE (SSC)
		. L - PRINTER DEVICE (SSC)
		. P - PROCESSOR DEVICE (SPC)
		. .W - WRITE ONCE READ MULTIPLE DEVICE (SBC)
		. . R - CD DEVICE (MMC)
		. . S - SCANNER DEVICE (SGC)
		. . .O - OPTICAL MEMORY DEVICE (SBC)
		. . . M - MEDIA CHANGER DEVICE (SMC)
		. . . C - COMMUNICATION DEVICE (SSC)
		. . . .A - STORAGE ARRAY DEVICE (SCC)
		. . . . E - ENCLOSURE SERVICES DEVICE (SES)
1Ah	DTL WRSOMCA	Power Condition mode page
1Bh	A	LUN mapping mode page
1Ch	DTL WRSOMCAE	Informational exceptions control mode page
1Dh	M	Transport geometry parameters mode page
1Eh	M	Element address assignments mode page
1Fh	M	Device capabilities mode page
00h		Vendor-specific (does not require page format)
20h THROUGH 29h		Vendor-specific (page format required)
2Ah	DTL W SOMCAE	Vendor-specific (page format required)
2Ah	R	CD capabilities and mechanical status mode page
2Bh THROUGH 3Eh		Vendor-specific (page format required)

Device column key
blank = reserved
not blank = allowed

Notes:
[1] Page code 0Dh is reserved for use by devices conforming to approved document X3T9.2/91-014r6. However, 1Ah is the preferred SCSI-3 page code for the Power Condition mode page.

Annex C
(informative)

Vendor identification

This annex contains the list of SCSI-3 vendor identifications (see table C.1) as of the date of this document. The purpose of this list is to help avoid redundant usage of vendor identifications. Technical Committee T10 of Accredited Standards Committee NCITS maintains an informal list of vendor identifications currently in use. Please contact the chairman of T10 prior to using a new vendor identification to avoid conflicts.

The information in this annex was complete and accurate at the time of publication. However, the information is subject to change. Technical Committee T10 of Accredited Standards Committee NCITS maintains an electronic copy of this information on its world wide web site (http://www.symbios.com/x3t10). In the event that the T10 world wide web site is no longer active, access may be possible via the NCITS world wide web site (http://www.x3.org).

Table C.1 – Vendor identification list

ID	Organization
3M	3M Company
ACL	Automated Cartridge Librarys, Inc.
AcuLab	AcuLab, Inc. (Tulsa, OK)
ADAPTEC	Adaptec
ADSI	Adaptive Data Systems, Inc. (a Western Digital subsidiary)
ADTX	ADTX Co., Ltd.
AERONICS	Aeronics, Inc.
AGFA	AGFA
AMCODYNE	Amcodyne
ANAMATIC	Anamartic Limited (England)
ANCOT	ANCOT Corp.
ANRITSU	Anritsu Corporation
APPLE	Apple Computer, Inc.
ARCHIVE	Archive
ASACA	ASACA Corp.
ASPEN	Aspen Peripherals
AST	AST Research
ASTK	Alcatel STK A/S
AT&T	AT&T
ATARI	Atari Corporation
ATG CYG	ATG Cygnet Inc.
ATTO	ATTO Technology Inc.
ATX	Alphatronix
AVR	Advanced Vision Research
BALLARD	Ballard Synergy Corp.
BERGSWD	Berg Software Design
BEZIER	Bezier Systems, Inc.
BiT	BiT Microsystems
BoxHill	Box Hill Systems Corporation
BULL	Bull Peripherals Corp.
BUSLOGIC	BusLogic Inc.
CalComp	CalComp, A Lockheed Company
CALIPER	Caliper (California Peripheral Corp.)
CAST	Advanced Storage Tech
CDC	Control Data or MPI
CDP	Columbia Data Products

(continued)

Table C.1 – Vendor identification list (continued)

ID	Organization
CHEROKEE	Cherokee Data Systems
CHINON	Chinon
CIE&YED	YE Data, C.Itoh Electric Corp.
CIPHER	Cipher Data Products
Ciprico	Ciprico, Inc.
CIRRUSL	Cirrus Logic Inc.
CMD	CMD Technology Inc.
CNGR SFW	Congruent Software, Inc.
COGITO	Cogito
COMPAQ	Compaq Computer Corporation
COMPORT	Comport Corp.
COMPSIG	Computer Signal Corporation
CONNER	Conner Peripherals
CORE	Core International, Inc.
CPU TECH	CPU Technology, Inc.
CREO	Creo Products Inc.
CROSFLD	Crosfield Electronics
CROSSRDS	Crossroads Systems, Inc.
CSM, INC	Computer SM, Inc.
DATABOOK	Databook, Inc.
DATACOPY	Datacopy Corp.
DATAPT	Datapoint Corp.
DEC	Digital Equipment (Obsolete: New products use "Digital")
DELL	Dell Computer Corporation
DELPHI	Delphi Data Div. of Sparks Industries, Inc.
DENON	Denon/Nippon Columbia
DenOptix	DenOptix, Inc.
DEST	DEST Corp.
DGC	Data General Corp.
DIGIDATA	Digi-Data Corporation
Digital	Digital Equipment Corporation
DILOG	Distributed Logic Corp.
DISC	Document Imaging Systems Corp.
DPT	Distributed Processing Technology
DSI	Data Spectrum, Inc.
DSM	Deterner Steuerungs- und Maschinenbau GmbH & Co.
DTC QUME	Data Technology Qume
DXIMAGIN	DX Imaging
ECCS	ECCS, Inc.
ECMA	European Computer Manufacturers Association
Elms	Elms Systems Corporation
EMC	EMC Corp.
EMULEX	Emulex
EPSON	Epson
Eris/RSI	RSI Systems, Inc.
EXABYTE	Exabyte Corp.
FILENET	FileNet Corp.
FRAMDRV	FRAMEDRIVE Corp.
FUJI	Fuji Electric Co., Ltd. (Japan)
FUJITSU	Fujitsu
FUNAI	Funai Electric Co., Ltd.
FUTURED	Future Domain Corp.
Gen_Dyn	General Dynamics
GIGATAPE	GIGATAPE GmbH
GIGATRND	GigaTrend Incorporated
Goidelic	Goidelic Precision, Inc.
GOULD	Gould

(continued)

Table C.1 – Vendor identification list (continued)

ID	Organization
HITACHI	Hitachi America Ltd or Nissei Sangyo America Ltd
HONEYWEL	Honeywell Inc.
HP	Hewlett Packard
i-cubed	i-cubed ltd.
IBM	International Business Machines
ICL	ICL
IDE	International Data Engineering, Inc.
IGR	Intergraph Corp.
IMPLTD	Integrated Micro Products Ltd.
IMPRIMIS	Imprimis Technology Inc.
INITIO	Initio Corporation
INSITE	Insite Peripherals
INTEL	INTEL Corporation
IOC	I/O Concepts, Inc.
IOMEGA	Iomega
ISi	Information Storage inc.
ISO	International Standards Organization
ITC	International Tapetronics Corporation
JPC Inc.	JPC Inc.
JVC	JVC Information Products Co.
KENNEDY	Kennedy Company
KENWOOD	KENWOOD Corporation
KODAK	Eastman Kodak
KONAN	Konan
KONICA	Konica Japan
Kyocera	Kyocera Corporation
LAPINE	Lapine Technology
LASERDRV	LaserDrive Limited
LASERGR	Lasergraphics, Inc.
LION	Lion Optics Corporation
LMS	Laser Magnetic Storage International Company
MATSHITA	Matsushita
MaxOptix	Maxoptix Corp.
MAXSTRAT	Maximum Strategy, Inc.
MAXTOR	Maxtor Corp.
MDI	Micro Design International, Inc.
MEADE	Meade Instruments Corporation
MELA	Mitsubishi Electronics America
MELCO	Mitsubishi Electric (Japan)
MEMREL	Memrel Corporation
MEMTECH	MemTech Technology
MERIDATA	Oy Meridata Finland Ltd.
METRUM	Metrum, Inc.
MICROBTX	Microbotics Inc.
MICROP	Micropolis
MICROTEK	Microtek Storage Corp
Minitech	Minitech (UK) Limited
Minolta	Minolta Corporation
MINSCRIB	Miniscribe
MITSUMI	Mitsumi Electric Co., Ltd.
MOTOROLA	Motorola
MST	Morning Star Technologies, Inc.
MTNGATE	MountainGate Data Systems
NAI	North Atlantic Industries
NAKAMICH	Nakamichi Corporation
NatInst	National Instruments
NatSemi	National Semiconductor Corp.

(continued)

Table C.1 – Vendor identification list (continued)

ID	Organization
NCL	NCL America
NCR	NCR Corporation
NEC	NEC
NISCA	NISCA Inc.
NKK	NKK Corp.
NRC	Nakamichi Corporation
NSM	NSM Jukebox GmbH
NT	Northern Telecom
OAI	Optical Access International
OCE	Oce Graphics
OKI	OKI Electric Industry Co.,Ltd (Japan)
OMI	Optical Media International
OMNIS	OMNIS Company (FRANCE)
OPTIMEM	Cipher/Optimem
OPTOTECH	Optotech
ORCA	Orca Technology
OSI	Optical Storage International
OTL	OTL Engineering
PASCOsci	Pasco Scientific
PERTEC	Pertec Peripherals Corporation
PFTI	Performance Technology Inc.
PFU	PFU Limited
PICO	Packard Instrument Company
PIONEER	Pioneer Electronic Corp.
PLASMON	Plasmon Data
PRAIRIE	PrairieTek
PREPRESS	PrePRESS Solutions
PRESOFT	PreSoft Architects
PRESTON	Preston Scientific
PRIAM	Priam
PRIMAGFX	Primagraphics Ltd
PROCOM	Procom Technology
PTI	Peripheral Technology Inc.
QIC	Quarter-Inch Cartridge Drive Standards, Inc.
QUALSTAR	Qualstar
QUANTUM	Quantum Corp.
QUANTEL	Quantel Ltd.
R-BYTE	R-Byte, Inc.
RACALREC	Racal Recorders
RADSTONE	Radstone Technology
RGI	Raster Graphics, Inc.
RICOH	Ricoh
RODIME	Rodime
RTI	Reference Technology
SAMSUNG	Samsung Electronics Co., Ltd.
SANKYO	Sankyo Seiki
SANYO	SANYO Electric Co., Ltd.
SCREEN	Dainippon Screen Mfg. Co., Ltd.
SDI	Storage Dimensions, Inc.
SEAGATE	Seagate
SEQUOIA	Sequoia Advanced Technologies, Inc.
Shinko	Shinko Electric Co., Ltd.
SIEMENS	Siemens
SII	Seiko Instruments Inc.
SMS	Scientific Micro Systems/OMTI
SNYSIDE	Sunnyside Computing Inc.
SONIC	Sonic Solutions

(continued)

Table C.1 – Vendor identification list (continued)

ID	Organization
SONY	Sony Corporation Japan
SPECIAL	Special Computing Co.
SPECTRA	Spectra Logic, a Division of Western Automation Labs, Inc.
SPERRY	Sperry (now Unisys Corp.)
Sterling	Sterling Diagnostic Imaging, Inc.
STK	Storage Technology Corporation
STORM	Storm Technology, Inc.
StrmLgc	StreamLogic Corp.
SUMITOMO	Sumitomo Electric Industries, Ltd.
SUN	Sun Microsystems, Inc.
SYMBIOS	Symbios Logic Inc.
SyQuest	SyQuest Technology, Inc.
SYSGEN	Sysgen
T-MITTON	Transmitton England
TALARIS	Talaris Systems, Inc.
TALLGRAS	Tallgrass Technologies
TANDBERG	Tandberg Data A/S
TANDON	Tandon
TEAC	TEAC Japan
TECOLOTE	Tecolote Designs
TEGRA	Tegra Varityper
Tek	Tektronix
TENTIME	Laura Technologies, Inc.
TI-DSG	Texas Instruments
TOSHIBA	Toshiba Japan
ULTRA	UltraStor Corporation
UNISYS	Unisys
USCORE	Underscore, Inc.
USDC	US Design Corp.
VERBATIM	Verbatim Corporation
VEXCEL	VEXCEL IMAGING GmbH
VICOMSL1	Vicom Systems, Inc.
VRC	Vermont Research Corp.
WangDAT	WangDAT
WANGTEK	Wangtek
WDIGTL	Western Digital
WEARNES	Wearnes Technology Corporation
WSC0001	Wisecom, Inc.
X3	Accredited Standards Committee X3, Information Technology
XEBEC	Xebec Corporation