American National Standard

for Information Technology –

SCSI/ATA Translation - 2 (SAT-2)

INCITS 465-2010

Developed by

incits SM

Where IT all begins

Approved American National Standard

ANSI

INCITS 465-2010

American National Standard
for Information Technology –

SCSI/ATA Translation - 2
(SAT-2)

Secretariat

Information Technology Industry Council

Approved November 12, 2010

American National Standards Institute, Inc.

Abstract

This standard specifies a translation layer between SCSI and ATA protocols. This translation layer is used by storage controllers to emulate objects in a SCSI logical unit using an ATA device, providing capabilities defined by SCSI standards (e.g., the SCSI Block Commands (SBC-3) and SCSI Primary Commands (SPC-4) standards). For the purposes of this standard, ATA device capabilities are defined by ATA8-AAM, ATA8-ACS, ATA8-APT, ATA8-AST, and SATA-2.6.

American National Standard

Approval of an American National Standard requires review 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 judgement 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 a concerted effort be made towards their resolution.

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

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an 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CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time.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 holders of patents that may be required for the implementation of the standard disclose such patents to the publisher. However, neither the developers nor the publisher have undertaken a patent search in order to identify which, if any, patents may apply to this standard. As of the date of publication of this standard, following calls for the identification of patents that may be required for the implementation of the standard, notice of one or more such claims has been received. By publication of this standard, no position is taken with respect to the validity of this claim or of any rights in connection therewith. The known patent holder(s) has (have), however, filed a statement of willingness to grant a license under these rights on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. Details may be obtained from the publisher. No further patent search is conducted by the developer or publisher in respect to any standard it processes. No representation is made or implied that this is the only license that may be required to avoid infringement in the use of this standard.

Published by

American National Standards Institute, Inc.
25 West 43rd Street, New York, NY 10036

Printed in the United States of America

Contents

Tables

Figures

Foreword (This foreword is not part of American National Standard INCITS 465-2010.)

This standard provides a common set of definitions and requirements to establish common behavior among implementations that emulate SCSI device behavior through the combined use of ATA devices and a SCSI / ATA Translation layer (SATL). The SATL may reside in a host-based software or firmware, or it may reside in a separate component (e.g., a host bus adapter or external controller) with a separate processing unit to perform the translation. A SATL and ATA device combination may provide a functional subset of common SCSI capabilities. There is also a range of optional emulated SCSI capabilities that may be supported, depending on the capabilities of the SATL.

This standard defines SATL capabilities in terms of SCSI capabilities as defined by the applicable SCSI standards and working drafts, and defines the elements and use of ATA protocol to provide those SCSI capabilities and services in a consistent manner among SAT implementations that implement according to this standard.

Requests for interpretation, suggestions for improvement and addenda, or defect reports are welcome. They should be sent to the INCITS Secretariat, National Committee for Information Technology Standards, Information Technology Institute, 1101 K Street, NW, Suite 610, Washington, DC 20005.

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

Don Wright, Chair
Jennifer Garner, Secretary

Organization Represented	*Name of Representative*
Adobe Systems, Inc.	Scott Foshee
	Steve Zilles (Alt.)
AIM Global, Inc.	Dan Mullen
	Charles Biss (Alt.)
Apple Computer, Inc.	Kwok Lau
	Helene Workman (Alt.)
	David Singer (Alt.)
Distributed Managment Task Force	John Crandall
	Jeff Hilland (Alt.)
Electronic Industries Alliance	Edward Mikoski, Jr.
	Henry Cuschieri (Alt.)
EMC Corporation	Gary Robinson
Farance, Inc.	Frank Farance
	Timothy Schoechle (Alt.)
Google	Zaheda Bhorat
GS1 US	Ray Delnicki
	Frank Sharkey (Alt.)
	James Chronowski (Alt.)
	Mary Wilson (Alt.)
Hewlett-Packard Company	Karen Higginbottom
	Paul Jeran (Alt.)
IBM Corporation	Gerald Lane
	Robert Weir (Alt.)
IEEE	Bill Ash
	Terry DeCourcelle (Alt.)
	Jodie Haasz (Alt.)
	Bob Labelle (Alt.)

Organization Represented	*Name of Representative*
Intel	Philip Wennblom
	Grace Wei (Alt.)
	Steven Balogh (Alt.)
Lexmark International	Don Wright
	Dwight Lewis (Alt.)
	Paul Menard (Alt.)
Microsoft Corporation	Jim Hughes
	Dave Welsh (Alt.)
	Mark Ryland (Alt.)
	John Calhoun (Alt.)
National Institute of Standards & Technology	Michael Hogan
	Elaine Barker (Alt.)
	Dan Benigni (Alt.)
	Fernando Podio (Alt.)
	Teresa Schwarzhoff (Alt.)
	Wo Chang (Alt.)
Oracle Corporation	Donald R. Deutsch
	Jim Melton (Alt.)]
	Michael Kavanaugh (Alt.)
	Toshihiro Suzuki (Alt.)
	Jeff Mischkinsky (Alt.)
	Tony DiCenzo (Alt.)
	Eduardo Gutentag (Alt.)
Purdue University	Stephen Elliott
Storage Networking Industry Association (SNIA)	Gary Phillips
	Arnold Jones (Alt.)
	Dave Thiel (Alt.)
US Department of Defense	Jerry Smith
	Dennis Devera (Alt.)
	Dave Brown (Alt.)
	Leonard Levine (Alt.)
US Department of Homeland Security	Peter Shebell
	Gregg Piermarini (Alt.)

Technical Committee T10 on SCSI Storage Interfaces, which developed and reviewed this standard, had the following members:

John B. Lohmeyer, Chair
Mark Evans, Vice-Chair
Ralph O. Weber, Secretary

Organization Represented	*Name of Representative*
Amphenol Interconnect	Gregory McSorley
	Adrian Green (Alt.)
	Alex Persaud (Alt.)
	Michael Wingard (Alt.)
Applied Micro Circuits Corporation	Scott Furey
Brocade	David Peterson
	Scott Kipp (Alt.)
Dell, Inc.	Kevin Marks
EMC Corporation	Gary Robinson
	David Black (Alt.)
	Adrianus Djohan (Alt.)
	Sean Dolan (Alt.)
	Mickey Felton (Alt.)
	Robert Payne (Alt.)
Emulex	William Martin
	Robert Nixon (Alt.)
ENDL Texas	Ralph Weber
	I Dal Allan (Alt.)
FCI Electronics	Douglas Wagner

Organization Represented	*Name of Representative*
Finisar Corporation	David Freeman
	Chris Cicchetti (Alt.)
	Dominic Coupal (Alt.)
	Paul Gentieu (Alt.)
	Geoff Hibbert (Alt.)
	Monica Li (Alt.)
	Katharine Schmidtke (Alt.)
Foxconn Electronics	Elwood Parsons
Fujitsu Comp Prod of America	Mike Fitzpatrick
	Don Jeanette (Alt.)
	Osamu Kimura (Alt.)
General Dynamics	Nathan Hastad
	Tim Mackley (Alt.)
Hewlett-Packard Company	Rob Elliott
	Curtis Ballard (Alt.)
	Michael Banther (Alt.)
	Wayne Bellamy (Alt.)
	Steven Fairchild (Alt.)
	Barry Olawsky (Alt.)
	Christopher Williams (Alt.)
	Jeff Wolford (Alt.)
Hitachi Global Storage Technologies	Dan Colegrove
	Dan Reno (Alt.)
IBM Corporation	Kevin Butt
	Mark Andresen (Alt.)
	Ted Vojnovich (Alt.)
Intel Corporation	Mark Seidel
	Heath Seabourn (Alt.)
	Pak-Lung Seto (Alt.)
	Sarah Sharp (Alt.)
Kawasaki Microelectronics America Inc	Joel Silverman
KnowledgeTek Inc	Dennis Moore
	Hugh Curley (Alt.)
Lexar Media Inc	John Geldman
	Neal Galbo (Alt.)
LSI Corporation	John Lohmeyer
	Brad Besmer (Alt.)
	Brian Day (Alt.)
	Keith Holt (Alt.)
	Walt Hubis (Alt.)
	Steve Johnson (Alt.)
	Bernhard Laschinsky (Alt.)
	Harvey Newman (Alt.)
	George Penokie (Alt.)
	Robert Sheffield (Alt.)
Marvell Semiconductor Inc	David Geddes
	Jacky Chow (Alt.)
	Paul Wassenberg (Alt.)
Maxim Integrated Products	Gregory Tabor
	David Allen (Alt.)
	Mahbubul Bari (Alt.)
	Samuel Barnett (Alt.)
	Kevin Witt (Alt.)
Microsoft Corporation	Calvin Chen
	Mark Benedikt (Alt.)
	Robert Griswold (Alt.)
	Frank Shu (Alt.)
Molex Inc	Jay Neer
	Galen Fromm (Alt.)
NetApp	Frederick Knight
	Chris Fore (Alt.)
	Subhash Sankuratripati (Alt.)
NVidia Corporation	Mark Overby
	Andrew Currid (Alt.)

Organization Represented	*Name of Representative*
PMC-Sierra	Tim Symons
	Guillaume Fortin (Alt.)
	Mathieu Gagnon (Alt.)
	Rick Hernandez (Alt.)
	Neil Wanamaker (Alt.)
Quantum Corporation	Paul Suhler
	Paul Stone (Alt.)
	Rod Wideman (Alt.)
Quellan	Gourgen Oganessyan
Samsung Information Systems	Joseph Chen
	Edward Chang (Alt.)
	Sung Lee (Alt.)
	Dmitry Obukhov (Alt.)
	John Osterlund (Alt.)
SanDisk IL Ltd.	Avraham Shimor
	Dave Landsman (Alt.)
	Yoni Shternhell (Alt.)
Seagate Technology	Gerald Houlder
	Alvin Cox (Alt.)
	Jim Hatfield (Alt.)
	Judy Westby (Alt.)
Sun Microsystems Inc	Dale LaFollette
	Jon Allen (Alt.)
	Dennis Appleyard (Alt.)
	Matthew Ball (Alt.)
	Vit Novak (Alt.)
	Scott Painter (Alt.)
Symantec	Roger Cummings
	Raymond Gilson (Alt.)
Tyco Electronics	Scott Shuey
	Michael Fogg (Alt.)
	Dan Gorenc (Alt.)
	Tom Grzysiewicz (Alt.)
	John Hackman (Alt.)
	Andy Nowak (Alt.)
	Robert Wertz (Alt.)
Volex Inc	Atul Sharma
	Toney Chew (Alt.)
Western Digital Corporation	Mark Evans
	Michael Rogers (Alt.)
	Curtis Stevens (Alt.)

Introduction

The SCSI / ATA Translation - 2 (SAT-2) standard is divided into the following clauses:

Clause 1 defines the scope of this standard.
Clause 2 enumerates the normative references that apply to this standard.
Clause 3 describes the definitions, symbols, abbreviations, and notation conventions used in this standard.
Clause 4 describes the general framework for defining elements of translation between SCSI and ATA protocol.
Clause 5 describes elements of SCSI / ATA Translation that relate to the SCSI architecture model.
Clause 6 describes the mapping of command management functions in the SATL layer.
Clause 7 provide a summary of SCSI commands mapped to ATA in this standard.
Clause 8 describes the mapping between SCSI Primary Commands and ATA protocol.
Clause 9 describes the mapping between SCSI Block Commands and ATA protocol.
Clause 10 describes the mapping of mode pages, log pages, and VPD page information to selected ATA protocol elements.
Clause 11 describes error reporting and sense data conventions for SCSI / ATA Translation.
Clause 12 describes SCSI commands and mode pages to support SCSI / ATA Translation.
Annex A describes command translation for ATAPI devices.

AMERICAN NATIONAL STANDARD **INCITS 465-2010**

American National Standard
for Information Technology–

SCSI/ATA Translation - 2 (SAT-2)

1 Scope

The set of SCSI standards specifies the interfaces, functions, and operations necessary to ensure interoperability between conforming SCSI implementations. This standard is a functional description. Conforming implementations may employ any design technique that does not violate interoperability.

This standard defines the protocol requirements of the SCSI / ATA Translation Layer (SATL) to allow conforming SCSI / ATA translating components to interoperate with ATA devices and SCSI application layers. The SATL covers the range of implementations that use ATA devices to emulate the behavior of SCSI devices as viewed by the SCSI application layer. The primary focus of this standard is to define SCSI / ATA Translation for an ATA device (see 3.1.9).

Where possible, this standard defines SCSI / ATA Translation in a manner that is consistent with the SAM-4, SPC-4, and SBC-3 standards. In some instances, the defined function of an ATA device is different from corresponding functions defined for SCSI target devices (e.g., an ATA device provides no means to abort a single ATA queued command). The translation defined in this standard, in such cases, may not be consistent with other SCSI standards. However, in such cases, this standard specifies the expected behavior, and in what manner it is inconsistent with the behavior specified in other SCSI standards.

The objective of this standard is to allow a complete set of SCSI functions while minimizing the complexity of the SATL and preserving compatibility with existing SCSI application clients.

The objectives of the SATL are:

a) to provide host computers with device independence with respect to the ATA devices that have user storage capacity, and with respect to various implementations of the translation layer used to emulate the behavior of SCSI target devices;
b) to define common features and functions representing a subset of the capabilities available in SCSI devices that apply to SCSI / ATA Translation implementations;
c) to define common methods to manage aspects of ATA devices that do not map to previously defined features and functions of SCSI, with provision made for the addition of special features and functions; and
d) to provide consistent means for discovery and control of optional SCSI features that may or may not be emulated in SCS I / ATA translator implementations. These means are provided by specifying how transport specific features and functions are represented in a mixed-domain topology in a manner consistent with management of devices in a SCSI domain.

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

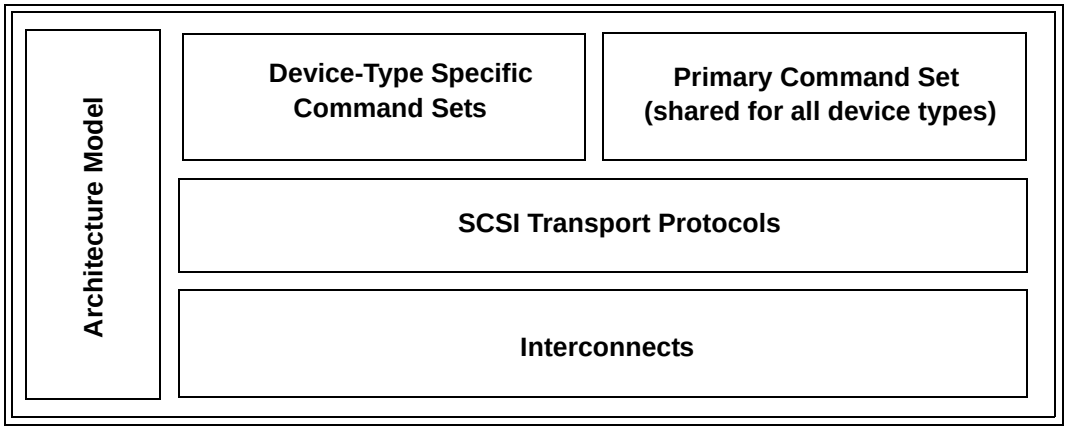


Figure 1 — SCSI document relationships

The term SCSI is used wherever it is not necessary to distinguish between the different SCSI standards.

Figure 2 shows the relationship of the ATA8 documents to each other.

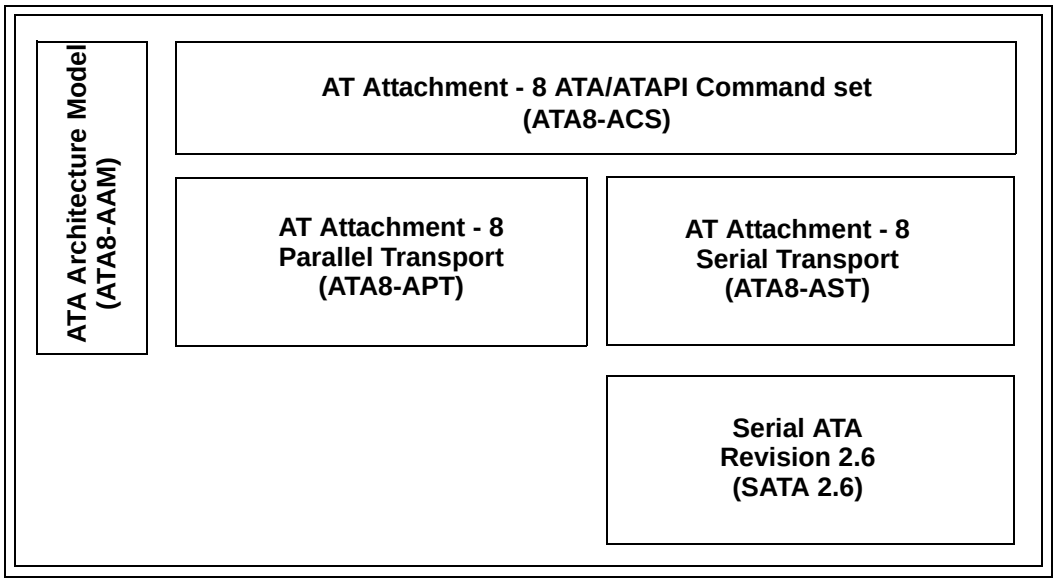


Figure 2 — ATA document structure

Figure 3 shows the relationship of this standard to standards in both the SCSI family of standards and the ATA family of standards.

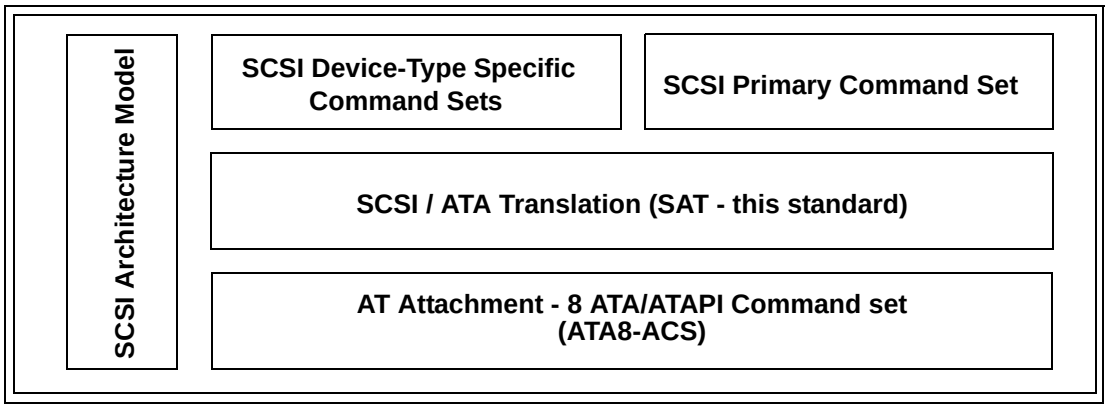


Figure 3 — SCSI / ATA Translation document role

This standard defines a translation between the SCSI application layer (see SAM-4) and ATA device protocol.

2 Normative References

2.1 Normative references

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

2.2 Approved references

Copies of the following documents may be obtained from ANSI, an ISO member organization:

a) Approved ANSI standards;
b) Approved international and regional standards (ISO and IEC); and
c) Approved foreign standards (including JIS and DIN).

For further information, contact the ANSI Customer Service Department:

Phone: +1 212-642-4900
Fax: +1 212-302-1286
Web: http://www.ansi.org
E-mail: ansionline@ansi.org

or the International Committee for Information Technology Standards (INCITS):

Phone: +1 202-626-5738
Web: http://www.incits.org
E-mail: incits@itic.org

ISO/IEC 14776-413, *SCSI Architecture Model - 3 (SAM-3)* [ANSI INCITS 402-2005]
ISO/IEC 14776-412, *SCSI Architecture Model - 2 (SAM-2)* [ANSI INCITS 366-2003]

2.3 References under development

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

AT Attachment-8 Architecture Model (ATA8-AAM) [T13/1700D]
AT Attachment-8 ATA/ATAPI Command Set (ATA8-ACS) [T13/1699D]
AT Attachment-8 Parallel Transport (ATA8-APT) [T13/1698D]
ISO/IEC 14776-454, *SCSI Primary Commands - 4 (SPC-4)* [T10/1731-D]
ISO/IEC 14776-323, *SCSI Block Commands - 3 (SBC-3)* [T10/1799-D]
ISO/IEC 14776-414, *SCSI Architecture Model - 4 (SAM-4)* [ANSI INCITS 447-2008]
ISO/IEC 14776-152, *Serial Attached SCSI - 2 (SAS-2)* [T10/1760-D]

2.4 Other references

For information on the current status of the listed document(s), or regarding availability, contact the indicated organization.

Serial ATA Revision 2.6 (SATA-2.6)

The SATA 2.6 document may be obtained from Serial ATA International Organization (SATA IO) at http://www.sata-io.org.

Mass Storage Class Bulk-Only Transport 1.0 (USB-BOT)

The USB-BOT document may be obtained from the USB Implementors Forum, Inc. at http://www.usb.org.

3 Definitions, symbols, abbreviations, and conventions

3.1 Definitions

3.1.1 additional sense code: A combination of the ADDITIONAL SENSE CODE field and the ADDITIONAL SENSE CODE QUALIFIER field in the sense data. See SPC-4.

3.1.2 Advanced Power Management (APM): The Advanced Power Management feature set as defined in ATA8-ACS.

3.1.3 allocation length: A value in the ALLOCATION LENGTH field of a CDB that specifies the maximum number of bytes that an application client has allocated in the Data-In Buffer and that is used to limit the maximum amount of variable length data (e.g., mode data, log data, diagnostic data) returned to an application client. See SPC-4.

3.1.4 application client: An object that is the source of SCSI commands. See SAM-4.

3.1.5 AT Attachment (ATA): A family of standards and specifications that define the attachment of storage devices to hosts. See ATA8-AAM, ATA8-ACS, ATA8-APT, and SATA-2.6.

3.1.6 AT Attachment Packet Interface (ATAPI): The PACKET Command feature set, as defined in ATA standards, that provides the capability to encapsulate SCSI and other types of commands and pass them over an ATA transport.

3.1.7 ATA abort retry: A policy implemented by a SATL whereby the SATL retries ATA commands aborted by ATA collateral abort (see 3.1.8) once.

3.1.8 ATA collateral abort: An ATA command that is aborted as a result of a different command being aborted when an ATA device is processing queued commands (i.e., NCQ or TCQ).

3.1.9 ATA device: A device compliant with ATA standards and implements the General feature set.

3.1.10 ATA device capacity: The ATA logical sector size, in bytes, (see 3.1.16) times one more than the ATA maximum LBA (see 3.1.17).

3.1.11 ATA domain: An I/O subsystem that is made up of one ATA host, a service delivery subsystem, and one or more ATA devices or ATAPI devices. See ATA8-AAM.

3.1.12 ATA flush command: A FLUSH CACHE command or a FLUSH CACHE EXT command. See ATA8-ACS.

3.1.13 ATA hardware reset: The routines performed by the ATA device server and the ATA device port in an ATA device after a hardware reset event occurs. See ATA8-AAM.

3.1.14 ATA host: An object that originates requests to be processed by an ATA device or an ATAPI device.

3.1.15 ATA LBA: A logical block address (see 3.1.49) used to reference a logical sector in an ATA device. See ATA8-ACS.

3.1.16 ATA logical sector size: The size of an ATA logical sector in bytes (see 5.7).

3.1.17 ATA maximum LBA: The maximum user LBA for the ATA device (see ATA8-ACS).

3.1.18 ATA nexus loss event: A transport-specific event where an ATA host port is no longer in communication with an ATA device port (see ATA8-AAM, see 5.5).

3.1.19 ATA non-queued command: An ATA command that is not an ATA queued command (see 3.1.20).

3.1.20 ATA queued command: A READ DMA QUEUED command, READ DMA QUEUED EXT command, WRITE DMA QUEUED command, WRITE DMA QUEUED EXT command, WRITE DMA QUEUED FUA EXT command, READ FPDMA QUEUED command, or WRITE FPDMA QUEUED command. See ATA8-ACS.

3.1.21 ATA read command: A READ DMA command, READ DMA EXT command, READ DMA QUEUED command, READ DMA QUEUED EXT command, READ MULTIPLE command, READ MULTIPLE EXT command, READ SECTOR(S) command, READ SECTOR(S) EXT command, or READ FPDMA QUEUED command. See ATA8-ACS.

3.1.22 ATA Sector Count: A count of ATA logical sectors to transfer or process, represented by the Count field in an ATA command. See ATA8-ACS.

3.1.23 ATA software reset: A reset that is triggered by an ATA task management function request (see ATA8-AAM, see 5.6).

3.1.24 ATA verify command: A READ VERIFY SECTOR(S) command or ATA READ VERIFY SECTOR(S) EXT command. See ATA8-ACS.

3.1.25 ATA volatile settings: ATA device settings affecting the way an ATA device responds to ATA commands that are configurable using ATA commands (e.g., ATA SET FEATURES command or ATA SET MAX EXT command), and that are set by the SATL to correspond to SCSI mode parameters, log parameters, or INQUIRY data.

3.1.26 ATA write command: A WRITE DMA command, WRITE DMA EXT command, WRITE DMA FUA EXT command, WRITE DMA QUEUED command, WRITE DMA QUEUED EXT command, WRITE DMA QUEUED FUA EXT command, WRITE MULTIPLE command, WRITE MULTIPLE EXT command, WRITE MULTIPLE FUA EXT command, WRITE SECTOR(S) command, WRITE SECTOR(S) EXT command, or WRITE FPDMA QUEUED command. See ATA8-ACS.

3.1.27 ATAPI device: A device that is compliant with the ATA standards and implements the PACKET feature set. See ATA8-ACS.

3.1.28 auto-contingent allegiance (ACA): The task set condition established following the return of a CHECK CONDITION status when the NACA bit is set to one in the CONTROL byte. See SAM-4.

3.1.29 autosense: Sense data that is returned in the same I_T_L_Q nexus transaction as the CHECK CONDITION status. See SAM-4. The alternative to autosense (i.e., use of a REQUEST SENSE command) is defined in SAM-2.

NOTE 1 - SAM-4 specifies what SAM-2 defines as autosense as the only valid way of returning SENSE data, but does not refer to it as autosense.

3.1.30 big-endian: A format for storage or transmission of binary data in which the most significant byte appears first. In a multi-byte value, the byte containing the most significant bit is stored in the lowest memory address and transmitted first and the byte containing the least significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 00h is stored in the lowest memory address and the byte containing 80h is stored in the highest memory address).

3.1.31 byte: A sequence of eight contiguous bits considered as a unit.

3.1.32 command: A request describing a unit of work to be performed by a device server. See SAM-4.

3.1.33 command descriptor block (CDB): A structure used to communicate a command from a SCSI application client to a SCSI device server.

3.1.34 device server: An object within the logical unit that processes SCSI commands according to the rules for command management. See SAM-4.

3.1.35 direct logical block mapping: A SATL implementation that maps logical blocks on a logical unit one-for-one with ATA logical sectors on an ATA device, where the LBA of a logical block has the same value as the LBA of the corresponding ATA logical sector and the number of bytes in a logical block equals the number of bytes in an ATA logical sector (see 9.1.2).

3.1.36 domain: A SCSI domain (see SAM-4) or an ATA domain (see ATA8-AAM).

3.1.37 DRQ data block: A unit of data words associated with available status when using either the PIO data-in command protocol or the PIO data-out command protocol. See ATA8-ACS

3.1.38 dword: A sequence of four contiguous bytes or four contiguous characters considered as a unit.

3.1.39 field: A group of one or more contiguous bits

3.1.40 indirect logical block mapping: A SATL implementation that does not follow the constraints of direct logical block mapping (see 3.1.35 and 9.1.3).

3.1.41 I_T nexus: A nexus between a SCSI initiator port and a SCSI target port. See SAM-4.

3.1.42 I_T_L nexus: A nexus between a SCSI initiator port, a SCSI target port, and a logical unit. See SAM-4.

3.1.43 I_T_L_Q nexus: A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a command. See SAM-4.

3.1.44 least significant bit (LSB): In a binary code, the bit or bit position with the smallest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 0001b, the bit that is set to one).

3.1.45 link reset: Performing the link reset sequence.

3.1.46 link reset sequence: A phy reset sequence. See SATA-2.6.

3.1.47 little-endian: A format for storage or transmission of binary data in which the least significant byte appears first. In a multi-byte value, the byte containing the least significant bit is stored in the lowest memory address and transmitted first and the byte containing the most significant bit is stored in the highest memory address and transmitted last (e.g., for the value 0080h, the byte containing 80h is stored in the lowest memory address and the byte containing 00h is stored in the highest memory address).

3.1.48 logical block: A set of data bytes accessed and referenced as a unit.

3.1.49 logical block address (LBA): The value used to reference a logical block.

3.1.50 logical unit: An externally addressable entity within a SCSI target device. See SAM-4 for a detailed definition of a logical unit.

3.1.51 logical unit capacity: The capacity of a logical unit in bytes calculated as length in bytes of each logical block times one more than the LBA of the last logical block on the logical unit.

3.1.52 logical unit number (LUN): An identifier for a logical unit. See SAM-4.

3.1.53 logical unit reset event: An event that triggers a logical unit reset. See SAM-4.

3.1.54 logical unit reset: A condition resulting from a hard reset condition or a logical unit reset event in which the logical unit performs the logical unit reset operations described in SAM-4, SPC-4, and this standard.

3.1.55 medium: The material on which data is stored (e.g., a magnetic disk).

3.1.56 most significant bit (MSB): In a binary code, the bit or bit position with the largest numerical weighting in a group of bits that, when taken as a whole, represent a numerical value (e.g., in the number 1000b, the bit that is set to one).

3.1.57 native command queuing (NCQ): A method by which a SATA device that does not implement the PACKET Command feature set may maintain and order the processing of up to 32 outstanding commands. See ATA8-ACS.

3.1.58 nexus: A relationship between a SCSI initiator port and a SCSI target port that may extend to a logical unit and a command. See SAM-4.

3.1.59 non-queued command: An ATA non-queued command (see 3.1.19).

3.1.60 object: An architectural abstraction or container that encapsulates data types, services, or other objects that are related in some way.

3.1.61 Parallel ATA (PATA): A parallel transport protocol. See ATA8-APT.

3.1.62 PATA bus: All of the conductors and connectors required to attain signal line continuity between every driver, receiver, and terminator for each signal between one PATA host and one or two PATA devices. See ATA8-APT.

3.1.63 PATA device: An ATA device or ATAPI device that uses the PATA transport protocol. See ATA8-APT.

3.1.64 PATA host: An ATA host that uses the PATA transport protocol. See ATA8-APT.

3.1.65 power on: Power being applied.

3.1.66 queued command: An ATA queued command (see 3.1.20), or a SCSI command received by the SATL from an application client for an emulated logical unit while the emulated logical unit is processing another SCSI command. See SAM-4.

3.1.67 reset event: A transport protocol specific event that results in a hard reset condition (see SAM-4) or a hardware reset (see ATA8-AAM).

3.1.68 SAS address: An identifier assigned to a SAS port or expander device. See SAS-2.

3.1.69 SAS initiator device: A device containing SSP, STP, and/or SMP initiator ports in a SAS domain. See SAS-2.

3.1.70 SAS initiator port: An SSP initiator port, STP initiator port, and/or SMP initiator port in a SAS domain. See SAS-2.

3.1.71 SATA device: An ATA device or ATAPI device that uses the Serial ATA transport protocol. See SATA-2.6.

3.1.72 SATA host: An ATA host that implements the Serial ATA transport protocol. See SATA-2.6.

3.1.73 SCSI / ATA Translation Layer (SATL): The functional layer defined in this standard that uses an ATA device to emulate objects in a SCSI logical unit, including the device server, task manager, and task set. See SAM-4.

3.1.74 SCSI device: A device that contains one or more SCSI ports that are connected to a service delivery subsystem and supports a SCSI application protocol.

3.1.75 SCSI hard reset: A condition resulting from a power on condition or a reset event in which the SCSI device performs the hard reset operations described in SAM-4, SPC-4, and the appropriate command and transport standards.

3.1.76 SCSI initiator port: A SCSI initiator device object that acts as the connection between application clients and a service delivery subsystem through which requests and responses are routed. See SAM-4.

3.1.77 SCSI read command: A READ (6), READ (10), READ (12), or READ (16) command. See SBC-3.

3.1.78 SCSI synchronize cache command: A SYNCHRONIZE CACHE(10) or SYNCHRONIZE CACHE (16) command. See SBC-3.

3.1.79 SCSI target port: A SCSI target device object that contains a task router and acts as the connection between device servers and task managers and a service delivery subsystem through which requests and responses are routed. See SAM-4.

3.1.80 SCSI verify command: A VERIFY (10), VERIFY (12), or VERIFY (16) command. See SBC-3.

3.1.81 SCSI write command: A WRITE (6), WRITE (10), WRITE (12), or WRITE (16) command. See SBC-3.

3.1.82 SCSI write and verify command: A WRITE AND VERIFY (10), WRITE AND VERIFY (12), or WRITE AND VERIFY (16) command. See SBC-3.

3.1.83 Serial ATA (SATA): A serial transport protocol that serves as an ATA service delivery subsystem. See SATA-2.6.

3.1.84 Serial ATA Tunneled Protocol (STP): The protocol used by STP initiator ports to communicate with STP target ports in a SAS domain. See SAS-2.

3.1.85 Serial Attached SCSI (SAS): A set of protocols and the interconnect defined by SAS-2.

3.1.86 service delivery subsystem: That part of a SCSI I/O system that transmits service requests to a logical unit or SCSI target device and returns logical unit or SCSI target device responses to a SCSI initiator device (see SAM-4) or that part of an ATA I/O system that connects an ATA host port and one or more ATA/ATAPI device ports and is a single path for the transfer of requests and responses between a host and one or more devices (see ATA8-AAM).

3.1.87 service response: The device service response or SCSI transport protocol specific service response returned to an application client by the SATL on completion of a SCSI transport protocol service request. See SAM-4.

3.1.88 STP initiator port: A SAS initiator device object in a SAS domain that interfaces to a service delivery subsystem with STP. See SAS-2.

3.1.89 STP target port: A SAS target device object in a SAS domain that interfaces to a service delivery subsystem with STP. See SAS-2.

3.1.90 STP/SATA bridge: An expander device object containing an STP target port, a SATA host port, and the functions required to forward information between the STP target port and SATA host port to enable STP initiator ports in a SAS domain to communicate with SATA devices in an ATA domain. See SAS-2.

3.1.91 task management function: A task manager service capable of being requested by an application client to affect the processing of one or more commands. See SAM-4.

3.1.92 task set: A group of commands within a device server whose interaction is dependent on the task management and auto-contingent allegiance rules. See SAM-4.

3.1.93 tagged command queuing (TCQ): A method that makes use of the ATA Tagged Command Queuing feature set, by which an ATA device may maintain and order the processing of up to 32 outstanding commands, identifying the context of each outstanding command with a unique tag. See ATA8-ACS.

3.1.94 Transport Protocol-Specific Information Unit (TPSIU): A transport-specific information unit used to transport information between initiator ports and target ports that may contain additional information needed by a service delivery subsystem to effect the requested information unit transfers (e.g., the Command Block Wrapper defined in USB-BOT).

3.1.95 word: A sequence of two contiguous bytes considered as a unit.

3.2 Symbols and abbreviations

≠ or NE	not equal
≤ or LE	less than or equal to
±	plus or minus
≈	approximately
x	multiply
+	add
-	subtract
< or LT	less than
= or EQ	equal
> or GT	greater than
≥ or GE	greater than or equal to
ACA	auto-contingent allegiance (see 3.1.28)
APM	Advanced Power Management (see 3.1.2)
ATA	AT Attachment (see 3.1.5)
ATAPI	AT Attachment Packet Interface (see 3.1.5)
CDB	Command Descriptor Block (see 3.1.33)
FIS	Frame Information Structure (see SATA-2.6)
FUA	force unit access
LBA	Logical Block Address (see 3.1.49)
LSB	least significant bit (see 3.1.44)
LUN	logical unit number (see 3.1.52)
MSB	most significant bit (see 3.1.56)
n/a	not applicable
NCQ	Native Command Queuing (see 3.1.57)
PATA	Parallel ATA (see 3.1.61)
SAS	Serial Attached SCSI (see 3.1.85)
SAT	SCSI / ATA Translation
SATA	Serial ATA (see 3.1.83)
SATA 2.6	Serial ATA-2.6 specification (see 2.4)
SATL	SCSI / ATA Translation Layer (see 3.1.73)
SAM-2	SCSI Architecture Model-2 standard (see 2.2)
SAM-3	SCSI Architecture Model-3 standard (see 2.2)
SAM-4	SCSI Architecture Model-4 standard (see 2.2)
SBC-3	SCSI Block Commands-3 standard (see 2.2)
SCSI	Small Computer System Interface family of standards
SCT	Smart Command Transport (see ATA8-ACS)
SPC-4	SCSI Primary Commands-4 standard (see 2.2)
STP	Serial ATA Tunneled Protocol (see 3.1.84)
SW	software
TCQ	Tagged Command Queuing (see 3.1.93)
TPSIU	Transport Protocol-Specific Information Unit (see 3.1.94)

VPD vital product data (see SPC-4)

3.3 Keywords

3.3.1 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 an error.

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

3.3.3 may: A keyword that indicates flexibility of choice with no implied preference.

3.3.4 may not: Keywords that indicates flexibility of choice with no implied preference.

3.3.5 obsolete: A keyword indicating that an item was defined in a previous version of a standard but has been removed from the most recent version of that 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. Their use and interpretation may be specified by future extensions to this or other standards. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this standard. Recipients are not required to check reserved bits, bytes, words or fields for zero values. Receipt of reserved code values in defined fields shall be reported as an error.

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

3.3.9 should: A keyword indicating flexibility of choice with a preferred alternative (equivalent to “it is strongly recommended”).

3.3.10 vendor specific: A keyword indicating specification of the referenced item is determined by the SCSI device vendor.

3.4 SAT specific terminology

3.4.1 emulated: A term designating that the SATL is required to implement functions in addition to or in place of functions supported by an ATA device to provide a defined SCSI capability.

3.4.2 unspecified: A term designating that this version of this standard does not specify a translation for a SCSI field. A translation for an unspecified field may be specified by future versions of this standard. Translation of fields marked unspecified shall not conflict with other standards in the set of SCSI standards.

3.5 Conventions

3.5.1 Overview

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, and additional sense codes are in all uppercase (e.g., REQUEST SENSE). Lowercase is used for words having the normal English meaning.

If there is more than one CDB length for a particular command (e.g., MODE SENSE (6) and MODE SENSE (10)) and the name of the command is used in a sentence without any CDB length descriptor (e.g., MODE SENSE), then the condition specified in the sentence applies to all CDB lengths for that command.

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

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.

3.5.2 Numeric conventions

A binary number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 and 1 immediately followed by a lower-case b (e.g., 0101b). Underscores or spaces may be included in binary number representations to increase readability or delineate field boundaries (e.g., 0 0101 1010b or 0_0101_1010b).

A hexadecimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 and/or the upper-case English letters A through F immediately followed by a lower-case h (e.g., FA23h). Underscores or spaces may be included in hexadecimal number representations to increase readability or delineate field boundaries (e.g., B FD8C FA23h or B_FD8C_FA23h).

A decimal number is represented in this standard by any sequence of digits consisting of only the Western-Arabic numerals 0 through 9 not immediately followed by a lower-case b or lower-case h (e.g., 25).

When the value of the bit or field is not relevant, x or xx appears in place of a specific value.

This standard uses the following convention for representing decimal numbers:

a) the decimal separator (i.e., separating the integer and fractional portions of the number) is a period;
b) the thousands separator (i.e., separating groups of three digits in the portion of a number) is a space; and
c) the thousands separator is used in both the integer and fractional portion of a number.

Table 1 shows some examples of decimal numbers using various conventions.

Table 1 — Numbering Conventions

French	English	This Standard
0,6	0.6	0.6
3,141 592 65	3.14159265	3.141 592 65
1 000	1,000	1 000
1 323 462,95	1,323,462.95	1 323 462.95

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

3.5.3 Bit and byte ordering

In this standard, data structures may be defined by a table. A table defines a complete ordering of elements (i.e., bits, bytes, fields, and dwords) within the structure. The ordering of elements within a table does not in itself constrain the order of storage or transmission of the data structure, but in combination with other normative text in this standard, may constrain the order of storage or transmission of the structure.

In a table, any element that is presented in a row above another element in a lower row is more significant than the lower element, and any element presented to the left of another element in the same row is more significant than the element to the right.

If a table shows bit numbering (see table 2), then the least significant bit (LSB) is numbered 0 and each more significant bit has the next greater number than the immediately less significant bit. If a table shows numbering of bytes or characters (see table 3), then the most significant byte or character is represented at the lowest number and each less significant byte or character has the next greater number than the immediately more significant byte.

In a field in a table consisting of more than one bit that contains a single value (e.g., a number), the least significant bit (LSB) is shown on the right and the most significant bit (MSB) is shown on the left (e.g., in a byte, bit 7 is the MSB and is shown on the left, bit 0 is the LSB and is shown on the right). The MSB and LSB are not labeled if the field consists of eight or fewer bits. The MSB and LSB are labeled if the field consists of more than eight bits and has no internal structure defined.

In a field in a table consisting of more than one byte that contains multiple fields each with their own values (e.g., a descriptor), there is no MSB and LSB of the field itself and thus there are no MSB and LSB labels. Each individual field has an MSB and LSB, but they are not labeled.

In a field containing a text string (e.g., ASCII or UTF-8), only the MSB of the first character and the LSB of the last character are labeled.

Multiple byte fields are represented with only two rows, with the non-sequentially increasing byte number denoting the presence of additional bytes.

A data dword consists of 32 bits. Table 2 shows a data dword containing a single value, where the MSB is on the upper left in bit 31 and the LSB is on the lower right in bit 0.

Table 2 — Example of ordering of bits and bytes within a multi-byte element

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) Bit 31	Bit 30	Bit 29	Bit 28	Bit 27	Bit 26	Bit 25	Bit 24
1	Bit 23	Bit 22	Bit 21	Bit 20	Bit 19	Bit 18	Bit 17	Bit 16
2	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8
3	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)

Table 3 shows a data dword containing four one-byte fields, where byte 0 (the first byte) is on the left and byte 3 (the fourth byte) is on the right. Each byte has an MSB on the left and an LSB on the right.

Table 3 — Example of ordering of bits and bytes within a multiple element

Bit / Byte	7	6	5	4	3	2	1	0
0	First byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
1	Second byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
2	Third byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)
3	Fourth byte							
	Bit 7 (MSB)	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0 (LSB)

3.5.4 Notation for byte encoded character strings

When this standard requires one or more bytes to contain specific encoded character, the specific characters are enclosed in single quotation marks. The single quotation marks identify the start and end of the characters that are required to be encoded but are not themselves to encoded. The characters that are to be encoded are shown in exactly the case that is to be encoded.

An ASCII space character (i.e., 20h) may be represented in a string by the character '¬' (e.g., 'SCSI¬device').

The encoded characters and the single quotation marks that enclose them are preceded by text that specifies the character encoding methodology and the number of characters required to be encoded.

The encoded characters and the single quotation marks that enclose them are preceded by text that specifies the character encoding methodology and the number of characters required to be encoded.

EXAMPLE - Using the notation described in this subclause, stating that eleven ASCII characters 'SCSI device' are to be encoded would be the same writing out the following sequence of byte values: 53h 43h 53h 49h 20h 64h 65h 76h 69h 63h 65h.

3.5.5 Notation for command descriptions

3.5.5.1 Description

The description of each command begins with a subclause describing the general method applied in translating the SCSI command to the corresponding ATA command(s), as well as any constraints and special considerations that may apply to the translation applied.

The subclause describing the general translation method for each command contains a table formatted like table 4 with two columns as follows:

a) the first column lists each of the fields in the SCSI CDB (see SPC-4 and SBC-3); and
b) the second column is either a brief description of the corresponding ATA features and functions used to implement the identified SCSI field, or a reference to a subsequent subclause containing a more lengthy description of the method of emulation or implementation.

Table 4 — Format for translated command field descriptions

Field	Description or reference
IMPLEMENTED OR EMULATED FIELD	A brief identification of the corresponding ATA features and functions, or a paragraph reference if there are special considerations that need to be applied in the use of the corresponding ATA features and functions that require a separate paragraph of description.
SUMMARY EMULATED FIELD	Summary field with more detailed structure.
UNSPECIFIED FIELD	Unspecified (see 3.4.2)

Tables listing fields in mode pages have an additional column that defines whether the field is changeable or not.

3.5.6 Use of field names defined in ATA standards and specifications

This standard discusses fields and values defined in other standards and specifications, in particular the ATA8-APT, ATA8-ACS, ATA8-AAM, and SCT standards developed by T13, and the SATA-2.6 specification. Such fields and values discussed in this standard are shown using the same notation conventions used in the standards where those fields and values are defined.

When this standard uses terms defined in T13 ATA standards or the SATA-2.6 specification, the following conventions apply:

a) The names of abbreviations, commands, and acronyms used as signal names are in all uppercase (e.g., IDENTIFY DEVICE). Fields containing only one bit are usually referred to as the “name” bit instead of the “name” field;
b) Names of device registers, fields in data structures, and other defined terms begin with an upper-case letter (e.g., Features field) and may be represented in mixed-case (e.g. PhyRdy);
c) The expression “word n” or “bit n” shall be interpreted as indicating the content of word n or bit n;
d) Bit names are shown in all uppercase letters; and
e) Bit (n:m) denotes a set of bits, for example, bits (7:0).

4 General

This standard defines a SCSI/ATA translation layer (i.e., the SATL) that provides a method for a SCSI application layer (see SAM-4) to access Serial ATA or Parallel ATA devices by representing ATA or ATAPI devices as SCSI peripheral devices.

Implementations of SCSI / ATA Translation may provide varying levels of SCSI functionality.

EXAMPLE 1 - The SATL may provide a level of SCSI emulation that is indistinguishable from native SCSI devices in terms of reported capabilities. Such SATL implementations need little guidance from this standard to effect interoperability since other SCSI protocol standards define all that is required to establish interoperability.

EXAMPLE 2 - The SATL may implement a subset of SCSI, have limited or no capability to maintain persistent information about the characteristics or state of the emulated SCSI device, have limited capability to manage device state information that carries forward from one command to the next, and maintain little or no capability to coordinate between multiple commands outstanding at a time. The characteristics and behavior of the underlying ATA devices in these minimal implementations of the SATL are expected to be more visible to the SCSI application clients.

This standard provides a set of definitions, conventions, and guidelines for:

a) the consistent reporting by the SATL of capabilities of emulated SCSI devices; and
b) the consistent identification of the attached devices by the application clients.

These provisions allow application clients to observe consistent behavior whether or not the application clients recognize the presence of a SATL in a system.

By defining expected behavior in terms of the SCSI commands received, corresponding activity in the ATA domain, and expected SCSI responses based on the results of activity in the ATA domain, this standard eliminates:

a) incompatibility between legacy SCSI / ATA Translation implementations; and
b) SCSI application client / ATA or ATAPI device interdependence.

This standard refers to behaviors for SCSI devices defined in SBC-3 and SPC-4. Unless otherwise specified, any behaviors that are optional in SBC-3 or SPC-4 are optional for devices implementing SCSI / ATA Translation. Any optional behaviors referred to in this standard and implemented by the SATL shall be implemented as described in this standard.

If the SATL receives a SCSI request specifying any value in any field of the CDB that the SATL does not support, unless otherwise specified in the description of the command, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB (see SPC-4).

If the SATL receives a SCSI request specifying any value in any field of the parameter data that the SATL does not support, unless otherwise specified in the description of the parameter, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST (see SPC-4).

5 SCSI architecture

5.1 Overview

This clause defines SCSI / ATA translation of features and functions that impact the representation of the domains defined in SAM-4 and ATA8-AAM. Figure 4 shows a SATL providing a communication path between a SCSI application client and an ATA or ATAPI device.

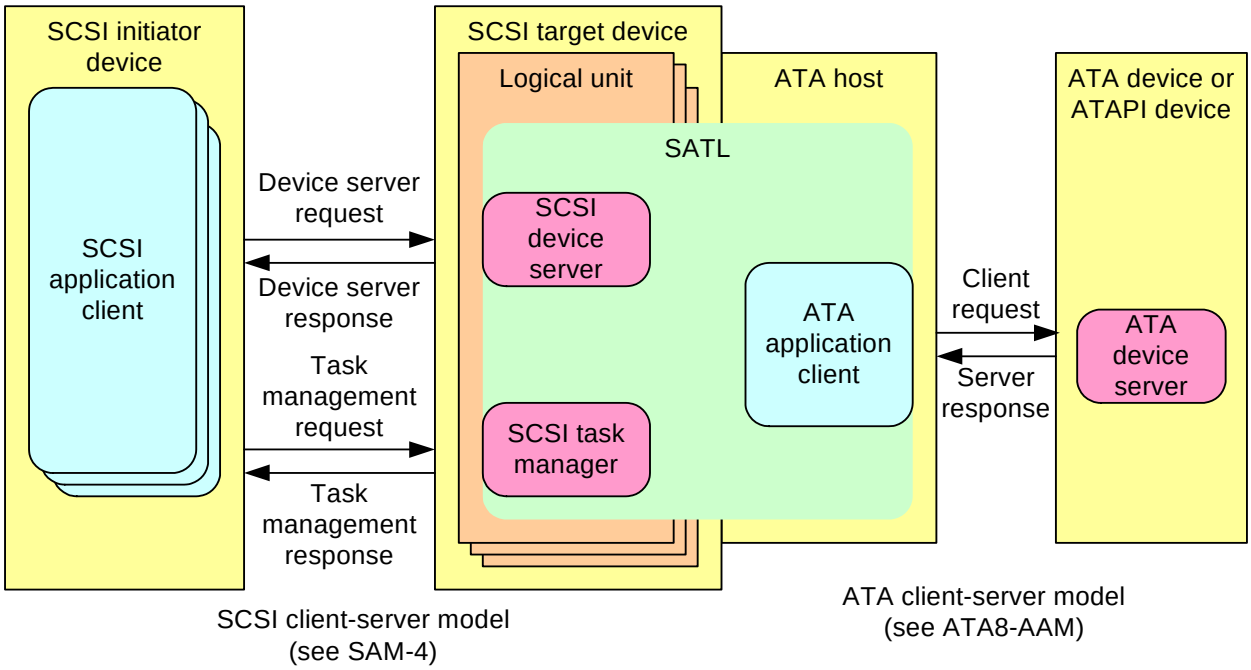


Figure 4 — Example of a SATL between a SCSI application client and an ATA or ATAPI device

The SATL provides the communication path between a SCSI application client and an ATA device or ATAPI device by:

a) emulating a SCSI logical unit;
b) integrating an ATA host; and
c) providing the translation that links them together.

This standard defines SCSI / ATA translation using SCSI and ATA command sets. This standard does not define the mapping of transport capabilities as defined at the SCSI transport protocol layer and the ATA protocol interconnect layer.

An implementation utilizing a SATL may include a SCSI transport. A SATL may appear in different configurations:

EXAMPLE 1 - Figure 5 shows a SATL contained within a SCSI to ATA protocol bridge, where the ATA or ATAPI device is being accessed by an ATA host port, and the SATL is being accessed with a SCSI target port using a SCSI transport protocol.

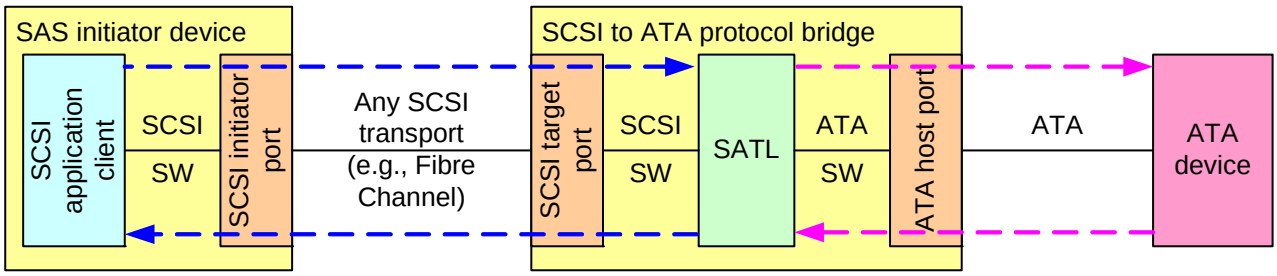


Figure 5 — SATL contained within a SCSI to ATA protocol bridge

EXAMPLE 2 - Figure 6 shows an ATA Host Bus Adapter (HBA) directly connected to an ATA device. The SATL provides SCSI transport protocol layer services to a SCSI application client in accordance with SAM-4.

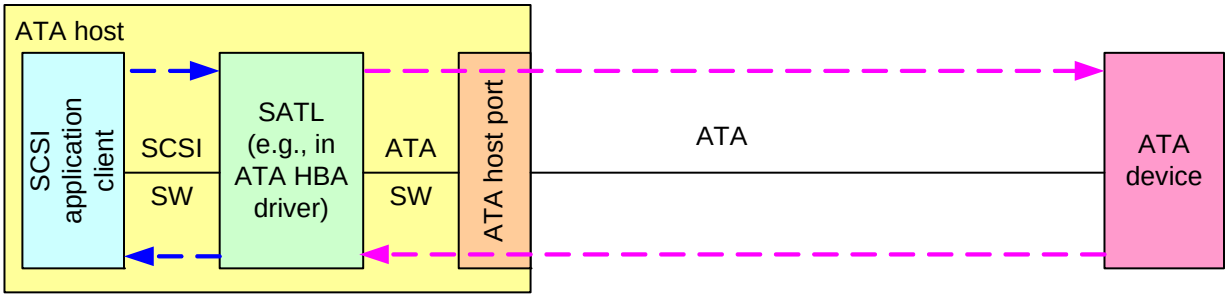


Figure 6 — SATL contained within an ATA host

EXAMPLE 3 - Figure 7 shows an ATA device accessed by a SAS STP initiator port (see SAS-2) through a SAS interconnect. The SAS initiator device includes a SATL to provide the SCSI transport protocol layer services to the application client in accordance with SAM-4.

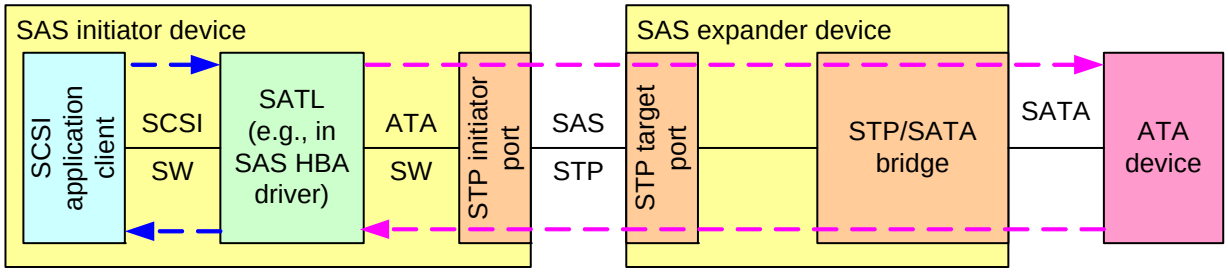


Figure 7 — SATL contained in a SAS initiator device

5.2 Multi-Initiator Configurations

SAM-4 defines configurations that may expose multiple I_T nexuses. Operation of a SATL exposed to multiple I_T nexuses are partially specified in this standard.

5.3 Unit attention condition

The SATL shall report events affecting the state of the emulated SCSI device to the SCSI application clients by emulating unit attention conditions (see SAM-4).

A SATL that detects a link reset for a Serial ATA attached ATA device or initiates any reset of an ATA device shall establish a unit attention condition on behalf of the logical unit corresponding to the ATA device with the sense key set to UNIT ATTENTION and the additional sense code set to POWER ON, RESET, OR BUS DEVICE RESET OCCURRED for the SCSI initiator port associated with each I_T nexus. The method a SATL uses to detect a link reset on the SATA link is vendor specific.

The SATL shall report unit attention conditions, in accordance with SAM-4, regardless of whether the condition results from accessing an ATA device or a condition internal to the SATL.

5.4 Handling errors in ATA commands

When a SCSI command is translated into one or more ATA commands and one of the ATA commands completes with an error, the SATL shall terminate processing of the SCSI command and report the error as described in clause 11.

When interpreting data from an ATA command, the SATL shall use the data only if no error was reported for the command. In addition:

a) when interpreting ATA IDENTIFY DEVICE data, if the Integrity word contains the Signature value defined in ATA8-ACS (i.e., word 255 bits 7:0), then the SATL shall use the data only if the Checksum is correct;
b) when interpreting ATA SMART READ DATA data for the ATA Summary SMART error log (i.e., log address 01h), the ATA Comprehensive SMART error log (i.e., log address 02h), the ATA SMART self-test log (i.e., log address 06h), or the ATA Selective self-test log (i.e., log address 09h) (see ATA8-ACS), the SATL shall use the data only if the data structure checksum (i.e., byte 511) is correct; and
c) when interpreting ATA READ LOG EXT data for the ATA Extended Comprehensive SMART error log (i.e., log address 03h) or ATA Extended SMART self-test log (i.e., log address 07h) (see ATA8-ACS), the SATL shall use the data only if the data structure checksum (i.e., byte 511) is correct.

5.5 ATA nexus loss

An ATA nexus loss event occurs when the SATL loses communication with the SATA device. If an ATA nexus loss event occurs, then:

a) the SATL shall terminate all commands being processed for the corresponding logical unit; and
b) the SATL shall establish a unit attention condition for each I_T nexus with the additional sense code set to:
    A) if the SATL is able to determine that the ATA device is no longer physically present, REPORTED LUNS DATA HAS CHANGED;
    B) if the SATL is unable to determine if the ATA device is physically present or not, INQUIRY DATA HAS CHANGED; or
    C) if the SATL is able to determine that the ATA device is present, INTERNAL TARGET FAILURE.

The method by which the SATL determines physical presence or absence of the ATA device is outside the scope of this standard (e.g., using cold presence detect (see SATA-2.6) or a change in the ELEMENT STATUS CODE field in the Device or Array Device element (see SES-2)).

NOTE 2 - SAM-4 and SPC-4 define how the SATL processes subsequent commands when the logical unit is no longer available (i.e., incorrect logical unit selection).

If the ATA nexus is restored or the SATL detects a power-on condition for an ATA device, then the SATL shall perform the processing described in 5.6 for those events.

5.6 ATA hardware and software reset processing

The hardware reset routines performed by the ATA device include the actions performed by the ATA device for an ATA software reset (see ATA8-AAM), the actions defined in ATA8-ACS, and the applicable ATA transport standards.

An ATA hardware reset may be caused either by the SATL or by the ATA device. If an ATA hardware reset or an ATA software reset occurs except as part of processing a SCSI task management function (see 6.3), then the SATL shall:

a) terminate processing of all commands for each logical unit affected by the reset;
b) restore the ATA volatile settings (see 3.1.25) of the ATA device (e.g., by sending an ATA SET FEATURES command) to values consistent with the saved values of mode pages if savable mode pages are supported and available, or default values if savable mode pages are not supported or are not available; and
c) establish a unit attention condition for each I_T_L nexus with the additional sense code set to POWER ON, RESET, OR BUS DEVICE RESET OCCURRED.

5.7 Translation of Large Physical Sectors

For SCSI large physical sector operation, see SBC-3 for information on the:

a) Logical Blocks model;
b) Physical Blocks model; and
c) READ CAPACITY(16) command.

For ATA large physical sector operation, see ATA8-ACS for information on the:

a) Long Logical Sector (LLS) feature set;
b) Long Physical Sector (LPS) feature set;
c) IDENTIFY DEVICE command;
d) Design and programming considerations for large physical sector devices annex; and
e) Implementation Guidelines for 1,024/4,096 Byte Sector Size Devices annex.

Table 5 describes parameters used in the translation and operation of large physical sectors and where the values for those parameters are found in both SCSI and ATA environments.

Table 5 — Large Physical Block Geometry Parameters

Parameter	SCSI	ATA
Logical Sector Size	READ CAPACITY (16) parameter data LOGICAL BLOCK LENGTH IN BYTES field	ATA IDENTIFY DEVICE data words 117 to 118
Logical Sectors Per Physical Sector Exponent	READ CAPACITY (16) parameter data LOGICAL BLOCKS PER PHYSICAL BLOCK field	ATA IDENTIFY DEVICE data words 106, bits 3:0
Logical Sectors Per Physical Sector	$2^{\text{SCSI Logical Sectors Per Physical Sector Exponent}}$	$2^{\text{ATA Logical Sectors Per Physical Sector Exponent}}$
Logical Sector Alignment	READ CAPACITY (16) parameter data LOWEST ALIGNED LOGICAL BLOCK ADDRESS field	ATA IDENTIFY DEVICE data word 209

Further relationships between SCSI and ATA follow:

a) SCSI Logical Sector Size is measured in bytes, whereas ATA Logical Sector size is measured in words;

b) ATA IDENTIFY DEVICE provides details on when the data contained in words 106, 117 through 118, and 209 are valid; and
c) The relationship between the SCSI and ATA logical sector alignment is:

SCSI Logical Sector Alignment = (y - ATA Logical Sector Alignment) modulus y

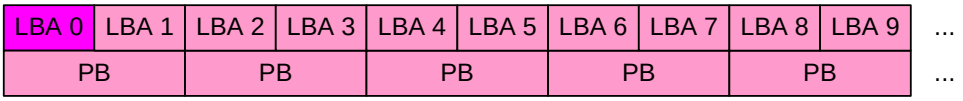
where
y = Logical Sectors Per Physical Sector

Figure 8, figure 9, and figure 10 show examples of physical to logical sector mapping.

Figure 8 — Logical Sector Alignment Examples (part 1 of 3)

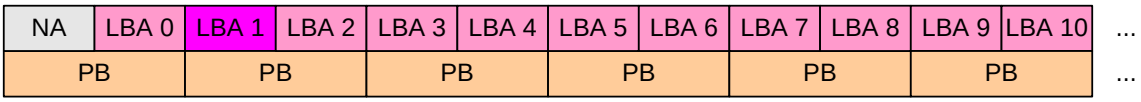
ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 1h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 1h
(indicating 2^1 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 1h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 1h:



ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 1h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 1h
(indicating 2^1 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 1h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 1h:



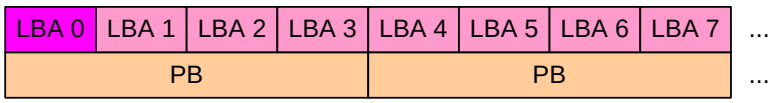
Key:
LBA n = logical block with LBA n
PB = physical block
NA= not accessible or addressable

The LOGICAL BLOCKS PER PHYSICAL BLOCK field and LOWEST ALIGNED LOGICAL BLOCK ADDRESS field are in the READ CAPACITY (16) command data.

Figure 9 — Logical Sector Alignment Examples (part 2 of 3)

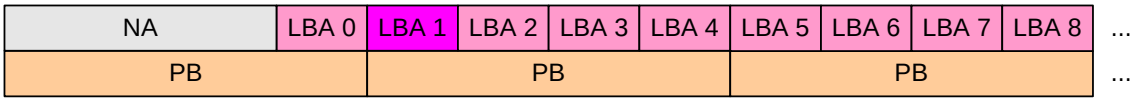
ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 2h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 2h
(indicating 2^2 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 0:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 0:



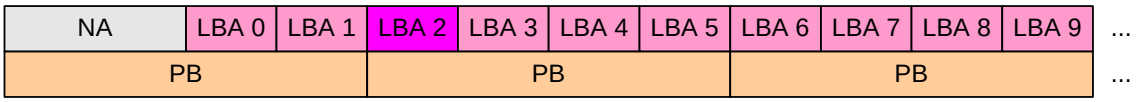
ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 2h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 2h
(indicating 2^2 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 3h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 1h:



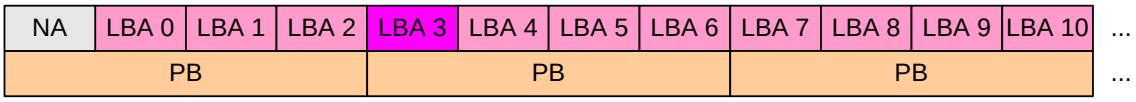
ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 2h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 2h
(indicating 2^2 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 2h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 2h:



ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 2h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 2h
(indicating 2^2 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 1h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 3h:



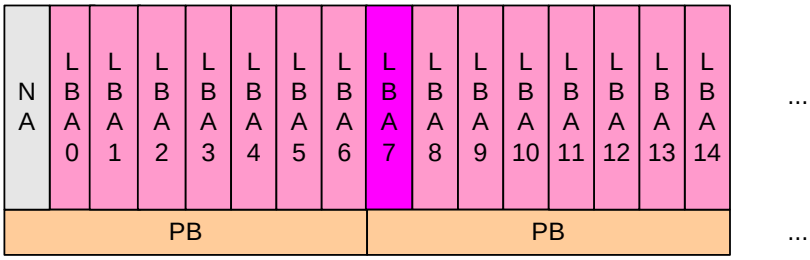
Key:
LBA n = logical block with LBA n
PB = physical block
NA = not accessible and not addressable

The LOGICAL BLOCKS PER PHYSICAL BLOCK field and LOWEST ALIGNED LOGICAL BLOCK ADDRESS field are in the READ CAPACITY (16) command data.

Figure 10 — Logical Sector Alignment Examples (Part 3 of 3)

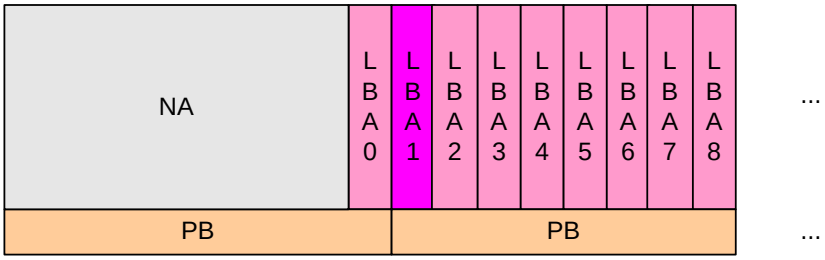
ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 3h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 3h
(indicating 2^3 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 1h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 7h:



ATA: LOGICAL SECTORS PER PHYSICAL SECTOR field set to 1h
SCSI: LOGICAL BLOCKS PER PHYSICAL BLOCK field set to 1h
(indicating 2^3 logical blocks per physical block):

ATA: LOGICAL SECTOR ALIGNMENT field set to 7h:
SCSI: LOWEST ALIGNED LOGICAL BLOCK ADDRESS field set to 1h:



Key:
LBA n = logical block with LBA n
PB = physical block
NA= not accessible or addressable

The LOGICAL BLOCKS PER PHYSICAL BLOCK field and LOWEST ALIGNED LOGICAL BLOCK ADDRESS field are in the READ CAPACITY (16) command data.

6 Command management model

6.1 Overview

A SATL may support the full task management model or the basic task management model as well as specific features of the task management model (e.g. SIMPLE and ORDERED task attributes) depending on the task management capabilities of the SATL and whether the SATL NCQ or TCQ.

6.2 Multiple command processing

6.2.1 Comparison of SCSI task set management and ATA queuing

Examples of the differences between SCSI task set management and ATA queuing methods are shown in table 6.

Table 6 — Comparison of SCSI task set management and ATA queuing methods

Feature [a]	SCSI	NCQ	TCQ
Ordering	Specified by task attributes (e.g. SIMPLE, ORDERED) associated with each command.	Always at the discretion of the device.	Always at the discretion of the device.
Queue Depth	Indeterminate	Fixed at 1 to 32 commands as reported in the ATA IDENTIFY DEVICE data word 75.	Fixed at 1 to 32 commands as reported in the ATA IDENTIFY DEVICE data word 75.
Queue full reporting	TASK SET FULL status	n/a	n/a
Queue full management	Device manages and indicates via TASK SET FULL status.	ATA host managed.	ATA host managed.
Queued commands	Task set management is applicable to all commands.	Limited to READ FPDMA QUEUED and WRITE FPDMA QUEUED commands.	Limited to READ DMA QUEUED, READ DMA QUEUED EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT and WRITE DMA QUEUED FUA EXT commands, or a NOP command with a non-zero subcommand code.
Handling of non-queued commands received while one or more queued commands are being processed	n/a	Receipt of any command other than a READ FPDMA QUEUED command or a WRITE FPDMA QUEUED command is an error.	Receipt of any command other than a NOP command with a non-zero subcommand code, a SERVICE command, a READ DMA QUEUED command, a READ DMA QUEUED EXT command, a WRITE DMA QUEUED command, a WRITE DMA QUEUED EXT command, or a WRITE DMA QUEUED FUA EXT command is an error.
Error handling	Controlled with mode parameters.	Any error aborts all queued commands.	Any error aborts all queued commands.

[a] Queue is a term used to represent a SCSI task set or an ATA queue.

6.2.2 Command mapping overview

A SATL that translates SCSI commands to an ATA device using NCQ or TCQ should implement the SAM-4 task management functions. If the SAM-4 task management functions are not implemented, then the SATL shall implement the basic task management model from SAM-3.

The SATL may implement internal queuing regardless of the version of the SCSI architecture model implemented.

6.2.3 Mapping of SCSI commands to ATA queued commands

A SATL that translates SCSI commands to an ATA device using NCQ or TCQ, whether or not the SATL also queues commands internally, shall either:

a) indicate support for the basic task management model in standard INQUIRY data (i.e., the BQUE bit is set to one and CMDQUE bit is set to zero), and follow the rules for the basic task management model (see SAM-3); or
b) indicate support for the full task management model in standard INQUIRY data (i.e., the BQUE bit is set to zero and CMDQUE bit is set to one), and set the QERR (i.e., queue error management) field of the Control mode page (see 10.1.5) as follows:
   A) a value of 01b if the SATL does not resend ATA queued commands aborted by the ATA device due to an error condition on any one of the ATA queued commands; or
   B) a value other than 01b if the SATL resends all other ATA queued commands (i.e., except the one in error) aborted by the ATA device due to an error condition on any one of the ATA queued commands.

For each SCSI command that the SATL translates to ATA queued commands (see 3.1.20), the SATL shall allocate an available tag value (e.g., for NCQ, the value corresponding to the position of a bit set to zero in the SActive register). The SATL shall maintain a mapping between allocated ATA queued command tags and the corresponding SCSI command identifier. See SAM-4.

The SATL should use the maximum queue depth supported by the ATA device (i.e., indicated by ATA IDENTIFY DEVICE data word 75), and may either:

a) return a status of TASK SET FULL in response to a SCSI command sent to the corresponding emulated SCSI logical unit when the ATA device represented has the maximum number of ATA queued commands outstanding; or
b) queue the SCSI command and return TASK SET FULL status when the SATL exhausts internal queueing resources.

6.2.4 Commands the SATL queues internally

If the translation of a SCSI command requires the SATL to send a non-queued command to the ATA device, then the SATL shall:

a) suspend processing of the SCSI command, maintain the SCSI command in a task set, and resume processing when the ATA device returns command complete for all ATA commands the SATL has previously sent to the ATA device;
b) return TASK SET FULL status for the SCSI command; or
c) return BUSY status for the SCSI command.

The SATL shall perform task management in accordance with the task management model (see SAM-3 or SAM-4) indicated in standard INQUIRY data and the Control mode page (see SPC-4).

6.2.5 Command queuing with multiple I_T nexuses

In some configurations the SATL may receive SCSI requests from multiple I_T nexuses. If the SATL receives SCSI requests from multiple I_T nexuses (e.g., the configuration shown in figure 5), as specified in SAM-4, the command identifiers maintained in the SATL mapping of command identifiers to NCQ tags or TCQ tags shall be qualified by the I_T nexus from which the command was received. When translating from an NCQ tag or TCQ tag to the corresponding SCSI command identifier, the SATL shall determine the correct I_T nexus using the qualification information associated with the SCSI command identifier. The SATL may return TASK SET

FULL status even if the ATA device has available NCQ tags or TCQ tags in order to maintain tags available for other I_T nexuses.

6.2.6 Collateral abort with queued commands

Error conditions with outstanding commands to an ATA device terminate all outstanding ATA commands being processed by the ATA device. An ATA host determines the status and error for each outstanding ATA queued command affected by the error condition and which ATA command(s) caused the error(s) (see ATA8-ACS or SATA-2.6). The SATL shall process aborted ATA commands as shown in table 7.

Table 7 — SATL processing of ATA commands aborted by ATA collateral abort

Association between the aborted ATA command and the ATA command that caused the error		Value of the QERR field set in the Control mode page	Method applied by the SATL for processing ATA commands aborted by ATA collateral abort
I_T_L_Q nexus	I_T nexus		
same		00b	The SATL shall terminate the command for the affected I_T_L_Q nexus with CHECK CONDITION status with the sense key and the additional sense code set according to the reported ATA error as described in clause 11.
		01b	
different	same	01b	The SATL shall terminate the affected I_T_L_Q nexus, but the SATL shall neither return status for the I_T_L_Q nexus affected by the aborted ATA command, nor retry the aborted ATA command.
		00b	The SATL shall resend the ATA command and continue processing the corresponding I_T_L_Q nexus.
different		00b	
		01b	The SATL shall terminate the affected I_T_L_Q nexus and establish a unit attention condition (see SAM-4) for the affected initiator port with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.

6.3 Command priority

A SATL that supports NCQ may also support SCSI command priority. SCSI command priority supports 16 priorities (0 to 15), whereas SATA NCQ only supports 2 priorities via the PRIO bit in the ATA READ FPDMA QUEUED command and ATA WRITE FPDMA QUEUED command. The SATL shall translate SCSI command priorities to SATA NCQ priority as shown in Table 8.

Table 8 — Command Priority to NCQ PRIO Mapping

SCSI command priority	SATA NCQ PRIO bit
0	0
1..3	1
4..15	0

6.4 Task management functions

6.4.1 Task management functions overview

This subclause describes the translation of SCSI task management functions to ATA equivalents.

See Annex A for task management function handling for ATAPI devices.

6.4.2 Aborting ATA queued commands

Some task management functions processed by the SATL may result in ATA commands aborted by ATA collateral abort (see 3.1.8) affecting an I_T_L_Q nexus other than the I_T_L_Q nexus(es) specified in the task management function request. 6.4 defines the translation for each task management function and defines how the SATL processes the I_T_L_Q nexuses affected by the task management function (e.g., ABORT TASK (see 6.4.4)).

Processing some task management functions requires the SATL to abort one or more ATA commands being processed by an ATA device.

The SATL shall abort an ATA queued command being processed by an ATA device by sending an ATA CHECK POWER MODE command or a READ LOG EXT (log page 10h) command to the ATA device.

NOTE 3 - The ATA CHECK POWER MODE command is used to abort ATA queued commands because it is an ATA non-queued command that does not transfer data. The ATA CHECK POWER MODE command does not affect ATA volatile settings.

6.4.3 Aborting ATA non-queued commands

To abort an ATA non-queued command the SATL shall:

a) send an ATA software reset to the ATA device; and
b) restore ATA volatile settings (see 3.1.25) to values consistent with current mode parameter settings.

6.4.4 ABORT TASK

The service request for the ABORT TASK task management function is (see SAM-4):

Service Response = ABORT TASK (IN (I_T_L_Q nexus)).

If no ATA commands associated with the I_T_L_Q nexus specified in the ABORT TASK task management function are outstanding to the ATA device, then the SATL shall abort the command for the specified I_T_L_Q nexus from the SATL internal context and respond to the ABORT TASK task management function with a service response of FUNCTION COMPLETE (see SAM-4).

If the ATA device is processing one or more ATA commands that are related to the specified I_T_L_Q nexus, then the SATL shall either:

a) allow the ATA command(s) to complete as follows:
   1) wait until the ATA device returns command complete for the ATA command(s);
   2) if the completed ATA command completes processing of the specified I_T_L_Q nexus, then return completion status for the I_T_L_Q nexus; and
   3) return a service response of FUNCTION COMPLETE for the ABORT TASK task management function regardless of whether or not completion status was returned for the I_T_L_Q nexus;

   or
b) abort the ATA command(s) (see 6.4.2) for the specified I_T_L_Q nexus and respond to the ABORT TASK task management function with a service response of FUNCTION COMPLETE.

If aborting the ATA commands related to the specified I_T_L_Q nexus results in one or more other ATA commands being aborted by ATA collateral abort (see 3.1.8), then the SATL shall:

a) if the SATL supports ATA abort retry (see 3.1.7), then re-send all ATA commands aborted by ATA collateral abort (see 3.1.8) and continue processing the affected I_T_L_Q nexuses; or
b) if the SATL does not support ATA abort retry, then for each I_T nexus affected by an ATA command aborted by ATA collateral abort:

1) terminate all but one of the SCSI commands without returning a function result; and
2) terminate processing of the remaining SCSI command by returning CHECK CONDITION status with the sense key set to UNIT ATTENTION and additional sense code set to COMMANDS CLEARED BY DEVICE SERVER.

6.4.5 ABORT TASK SET

The service request for the ABORT TASK SET task management function (see SAM-4) is:

Service Response = ABORT TASK SET (IN (I_T_L nexus)).

If the ATA device is not processing ATA commands for SCSI commands associated with the specified I_T_L nexus, then the SATL shall abort all commands for the specified I_T_L nexus from the SATL internal context and respond to the ABORT TASK SET task management function with a service response of FUNCTION COMPLETE.

If the ATA device is processing any ATA commands related to the specified I_T_L nexus, then the SATL shall either:

a) allow the ATA command(s) to complete as follows:
   1) wait until the ATA device returns command complete for the ATA command(s);
   2) if the completed ATA command completes processing a SCSI command in the task set, return completion status for the SCSI command; and
   3) after all ATA commands return completion status, return a service response of FUNCTION COMPLETE for the ABORT TASK SET task management function;

or

b) abort outstanding ATA command(s) (see 6.4.2) for the specified I_T_L nexus, and respond to the ABORT TASK SET task management function with a service response of FUNCTION COMPLETE.

If aborting ATA commands for the specified I_T_L nexus results in ATA commands aborted by ATA collateral abort (see 3.1.8) that are related to processing SCSI commands in an I_T_L nexus other than the specified I_T_L nexus, then:

a) if the SATL supports ATA abort retry (see 3.1.7), then the SATL shall re-send all ATA commands aborted by ATA collateral abort and continue processing of the affected I_T_L_Q nexuses; or
b) if the SATL does not support ATA abort retry, then for each I_T_L nexus other than the specified I_T_L nexus that had one or more SCSI commands affected due to ATA commands aborted by ATA collateral abort, the SATL shall abort all commands for each affected I_T_L nexus and establish a UNIT ATTENTION condition with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.

NOTE 4 - A SATL that does not support ATA abort retry (see 3.1.7) is not able to comply with the SAM-4 requirement that ABORT TASK SET not abort commands other than those in the specified I_T_L nexus.

6.4.6 CLEAR ACA

The service request for the CLEAR ACA task management function (see SAM-4) is:

Service Response = CLEAR ACA (IN (I_T_L nexus)).

The SATL shall respond to a CLEAR ACA task management function with a service response of FUNCTION REJECTED.

6.4.7 CLEAR TASK SET

The service request for the CLEAR TASK SET task management function (see SAM-4) is:

Service Response = CLEAR TASK SET (IN (I_T_L nexus)).

If the SATL indicates support for the full task management model (see 6.2.3), then the SATL shall process the CLEAR TASK SET task management function in accordance with a single task set that includes SCSI commands for all I_T_L nexuses (i.e., the TST field in the Control mode page is set to 000h, see 10.1.5).

If the ATA device is processing any ATA commands, then the SATL shall:

a) abort all outstanding ATA command(s);
b) abort all SCSI commands in the task set; and
c) respond to the CLEAR TASK SET task management function with a service response of FUNCTION COMPLETE.

If the SATL aborts commands in the task set for an I_T_L nexus other than the specified I_T_L nexus, then for each other I_T_L nexus, the SATL shall establish a unit attention condition with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.

6.4.8 I_T NEXUS RESET

The service request for the I_T NEXUS RESET task management function (see SAM-4) is:

Service Response = I_T NEXUS RESET (IN (I_T nexus)).

If the ATA device is not processing ATA commands for SCSI commands associated with the specified I_T nexus, then the SATL shall abort all commands for the specified I_T nexus from the SATL internal context and respond to the I_T NEXUS RESET task management function with a service response of FUNCTION COMPLETE.

If the ATA device is processing any ATA commands related to the specified I_T nexus, then the SATL shall either:

a) allow the ATA command(s) to complete as follows:
   1) wait until the ATA device returns command complete for the ATA command(s);
   2) if the completed ATA command completes processing a SCSI command in the task set, return completion status for the SCSI command; and
   3) after all ATA commands return completion status, return a service response of FUNCTION COMPLETE for the I_T NEXUS RESET task management function;

or

b) abort outstanding ATA command(s) (see 6.4.2) for the specified I_T nexus, and respond to the I_T NEXUS RESET task management function with a service response of FUNCTION COMPLETE.

The SATL shall establish a unit attention condition on behalf of the logical unit corresponding to the ATA device with an additional sense code set to I_T NEXUS LOSS OCCURRED.

If aborting ATA commands for the specified I_T nexus results in ATA commands being aborted by ATA collateral abort (see 3.1.8) that are related to processing SCSI commands in an I_T nexus other than the specified I_T nexus, then:

a) if the SATL supports ATA abort retry (see 3.1.7), then the SATL shall re-send all ATA commands aborted by ATA collateral abort and continue processing of the affected I_T_L_Q nexuses; or
b) if the SATL does not support ATA abort retry, then for each I_T nexus other than the specified I_T nexus that had one or more SCSI commands affected due to ATA commands aborted by ATA collateral abort, the SATL shall abort all commands for each affected I_T nexus and establish a UNIT ATTENTION condition with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR.

NOTE 5 - A SATL that does not support ATA abort retry (see 3.1.7) is not able to comply with the SAM-4 requirement that I_T NEXUS RESET not abort commands other than those in the specified I_T nexus.

If the SATL does not support the I_T NEXUS RESET task management function the SATL shall return a service response of FUNCTION REJECTED.

6.4.9 LOGICAL UNIT RESET

The service request for the LOGICAL UNIT RESET task management function (see SAM-4) is:

Service Response = LOGICAL UNIT RESET (IN (I_T_L nexus)).

The SATL shall:

1) reset the ATA device as follows:

1) optionally send an ATA software reset (see 3.1.23) to the ATA device; and
2) if the ATA software reset is not successful or not sent, then send an ATA hardware reset (see 3.1.13) to the ATA device;

NOTE 6 - It is vendor-specific how the SATL determines if the ATA software reset is successful.

2) abort all commands in the task set from the SATL internal context;
3) restore ATA volatile settings (see 3.1.25) to values consistent with the emulation of saved or default values of mode parameters, log parameters, and INQUIRY data (see SPC-4);
4) return a service response of FUNCTION COMPLETE for the LOGICAL UNIT RESET task management function; and
5) establish a unit attention condition (see SAM-4).

NOTE 7 - If more than one PATA device is present on a PATA bus, then issuing an ATA software reset causes both PATA devices to be reset.

6.4.10 QUERY TASK

The service request for the QUERY TASK task management function (see SAM-4) is:

Service Response = QUERY TASK (IN (I_T_L_Q nexus)).

If the SATL supports the QUERY TASK task management function, then the SATL shall return a service response of FUNCTION SUCCEEDED if the specified I_T_L_Q nexus is in the task set, or the SATL shall return a service response of FUNCTION COMPLETE if the specified I_T_L_Q nexus is not in the task set.

If the SATL does not support the QUERY TASK task management function then the SATL shall return a service response of FUNCTION REJECTED.

6.4.11 QUERY TASK SET

The service request for the QUERY TASK SET task management function (see SAM-4) is:

Service Response = QUERY TASK SET (IN (I_T_L nexus)).

If the SATL supports the QUERY TASK SET task management function, then:

a) the SATL shall return a service response of FUNCTION SUCCEEDED if there is any command present in the task set from the specified I_T_L nexus; or
b) the SATL shall return a service response of FUNCTION COMPLETE if there is no command present in the specified I_T_L nexus.

If the SATL does not support the QUERY TASK SET task management function the SATL shall return a service response of FUNCTION REJECTED.

6.4.12 QUERY ASYNCHRONOUS EVENT

The service request for the QUERY ASYNCHRONOUS EVENT task management function (see SAM-4) is:

Service Response = QUERY ASYNCHRONOUS EVENT (IN (I_T_L nexus), OUT ([Additional Response Information])).

If the SATL supports the QUERY ASYNCHRONOUS EVENT task management function, then the SATL shall:

a) if there is a unit attention condition or deferred error pending for the specified I_T nexus return a service response of FUNCTION SUCCEEDED; or
b) if there is a unit attention condition or deferred error pending for the specified I_T nexus return a service response of FUNCTION COMPLETE.

If the SATL supports the QUERY ASYNCHRONOUS EVENT task management function, then the SATL shall return the Additional Response Information as specified in SAM-4

If the SATL does not support the QUERY ASYNCHRONOUS EVENT task management function the SATL shall return a service response of FUNCTION REJECTED.

6.4.13 Reset task management functions

The TARGET RESET task management function (see SAM-2) is sometimes used by a SCSI application client to cause a hard reset (i.e., similar to a power-on condition) for each logical unit of a specified target device. The SATL may process the TARGET RESET task management function by issuing an ATA hardware reset (see 3.1.13) to the ATA device(s) associated with the target device.

6.5 CONTROL Byte

Table 9 describes SATL handling of the CDB CONTROL byte. See SAM-4 for CONTROL byte details.

Table 9 — CONTROL byte fields

Field	Description
Vendor specific	The SATL may use this field for vendor-specific purposes.
NACA	If set to one, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

6.6 I_T nexus loss

The SATL may detect an I_T nexus loss event (see SAM-4). If the SATL detects an I_T nexus loss event then the SATL handles the I_T nexus loss event differently depending on whether the SATL provides multiple I_T nexuses access to the emulated SCSI logical unit. See Annex A for I_T nexus loss handling for ATAPI devices.

If the SATL does not provide multiple I_T nexuses access to the emulated SCSI logical unit, then the SATL shall handle the I_T nexus loss as follows:

1) abort any outstanding ATA command(s) (see 6.4.2 and 6.4.3);
2) delete all commands in the task set from the SATL internal context; and
3) establish a unit attention condition for the affected I_T nexus with the additional sense code set to I_T NEXUS LOSS OCCURRED.

If the SATL provides multiple I_T nexuses access to the emulated SCSI logical unit, then the SATL shall handle the I_T nexus loss as follows:

1) allow any outstanding ATA command(s) for each I_T nexus that is not lost to complete;
2) abort any remaining ATA command(s) (see 6.4.2 and 6.4.3);
3) delete all commands in the task set from the SATL internal context for commands associated with the I_T nexus that the I_T nexus loss event occurred; and
4) establish a unit attention with the additional sense code set to I_T NEXUS LOSS OCCURRED for the SCSI initiator port associated with the I_T nexus that was lost.

7 Summary of SCSI / ATA command mappings

In the event of a discrepancy between the contents of this clause and the description of individual commands, description of individual commands shall apply.

Clause 7, clause 8, and clause 9 describe the SCSI to ATA command mapping for ATA devices emulating a SCSI logical unit with a peripheral device type of 00h (i.e., direct-access block device). Command transmission requirements for ATAPI devices are described in Annex A.

The SATL shall not send more than one ATA command to the ATA device representing the logical unit with the exception of ATA queued commands (see 3.1.20). The SATL shall queue received SCSI commands as necessary to enforce this.

Table 10 lists the SCSI / ATA command mappings defined in this standard. A SATL may implement commands defined in SPC-4 and SBC-3, but not listed in table 10. Translation of commands not listed in table 10 is vendor-specific.

Table 10 — Summary of SCSI / ATA Command Mapping (part 1 of 2)

SCSI command	ATA command(s)[a]	Reference
ATA PASS-THROUGH (12)	Any	12.2.2
ATA PASS-THROUGH (16)		12.2.3
FORMAT UNIT	READ VERIFY SECTORS, READ VERIFY SECTORS EXT, WRITE SECTORS, or WRITE SECTORS EXT	9.2
INQUIRY	IDENTIFY DEVICE	8.1
LOG SENSE	Log page dependent (see 10.2)	8.2
MODE SELECT (6)	Mode page dependent (see 10.1)	8.4
MODE SELECT (10)		8.5
MODE SENSE (6)		8.6
MODE SENSE (10)		8.7
READ (6)	See 9.1	9.3
READ (10)		9.5
READ (12)		9.6
READ (16)		9.7
READ BUFFER	READ BUFFER	8.8
READ CAPACITY (10)	IDENTIFY DEVICE	9.8
READ CAPACITY (16)		9.9
REASSIGN BLOCKS	READ VERIFY SECTOR(S), READ VERIFY SECTOR(S) EXT, WRITE DMA, WRITE DMA EXT, WRITE DMA FUA EXT, WRITE DMA QUEUED, WRITE DMA QUEUED EXT, WRITE DMA QUEUED FUA EXT, or WRITE FPDMA QUEUED	9.10
REPORT LUNS	n/a	SPC-4
REQUEST SENSE	SMART RETURN STATUS and CHECK POWER MODE	8.9
SECURITY PROTOCOL IN	TRUSTED RECEIVE, TRUSTED RECEIVE DMA, or TRUSTED NON-DATA	8.10
SECURITY PROTOCOL OUT	TRUSTED SEND, TRUSTED SEND DMA, or TRUSTED NON-DATA	8.11

[a] Translations for SCSI commands may require one or more of the ATA commands listed to be sent to the ATA device

Table 10 — Summary of SCSI / ATA Command Mapping (part 2 of 2)

SCSI command	ATA command(s)[a]	Reference
SEND DIAGNOSTIC	SMART EXECUTE OFF-LINE IMMEDIATE	8.12
START STOP UNIT	FLUSH CACHE, FLUSH CACHE EXT, STANDBY, IDLE IMMEDIATE, READ VERIFY SECTOR(S), READ VERIFY SECTOR(S) EXT, ATA verify commands, ATA flush commands, or STANDBY IMMEDIATE	9.11
SYNCHRONIZE CACHE (10)	FLUSH CACHE or FLUSH CACHE EXT	9.12
SYNCHRONIZE CACHE (16)		9.13
TEST UNIT READY	GET MEDIA STATUS and CHECK POWER MODE	8.13
VERIFY (10)	See 9.1	9.14
VERIFY (12)		9.15
VERIFY (16)		9.16
WRITE (6)	See 9.1	9.18
WRITE (10)		9.19
WRITE (12)		9.20
WRITE (16)		9.21
WRITE AND VERIFY (10)	See 9.1	9.23
WRITE AND VERIFY (12)		9.24
WRITE AND VERIFY (16)		9.25
WRITE BUFFER	WRITE BUFFER or DOWNLOAD MICROCODE	8.14
WRITE LONG (10)	WRITE UNCORRECTABLE EXT	9.26
WRITE LONG (16)		9.27
WRITE SAME (10)	See 9.1	9.28
WRITE SAME (16)		9.29

[a] Translations for SCSI commands may require one or more of the ATA commands listed to be sent to the ATA device

8 SCSI Primary Commands (SPC) command mapping

8.1 INQUIRY command

8.1.1 INQUIRY command overview

The INQUIRY command requests general information about a logical unit and target device. The INQUIRY command and selected VPD pages shall be emulated using information from the ATA IDENTIFY DEVICE command and other information (see 8.1.2). Table 11 describes the emulation of fields in the INQUIRY CDB.

Table 11 — INQUIRY CDB field translations

Field	Description or reference
OPERATION CODE	Set to 12h. The SATL shall send an ATA IDENTIFY DEVICE command to the ATA device.
EVPD	The SATL shall implement this field as defined in SPC-4 (see 10.3).
PAGE CODE[a]	The SATL: a) shall support the Supported VPD Pages VPD page (00h) (see 10.3.2); b) may support the Unit Serial Number VPD page (80h) (see 10.3.3); c) shall support the Device Identification VPD page (83h) (see 10.3.4); d) should support the Mode Page Policy VPD page (87h) (see 10.3.5); e) shall support the ATA Information VPD page (89h) (see 12.4.2); and f) may support the Block Device Characteristics VPD page (B1h) (see 10.3.6).
ALLOCATION LENGTH	The SATL shall implement this field as defined in SPC-4.
CONTROL	6.5

[a] VPD page translations are defined in 10.3.

8.1.2 Standard INQUIRY data

Table 12 describes the standard INQUIRY data fields supported by the SATL.

Table 12 — Standard INQUIRY data fields (part 1 of 3)

Field	Description or reference
PERIPHERAL QUALIFIER	The SATL shall set this field to 000b to indicate that the peripheral device is currently connected to this logical unit.[a]
PERIPHERAL DEVICE TYPE	The SATL shall set this field to 00h to indicate that the peripheral device is a direct access block device.[a]
RMB	The SATL shall set this bit to the value of bit 7 of the general configuration word of the ATA IDENTIFY DEVICE data retrieved from the ATA device.
VERSION	The VERSION field indicates the version of SPC to which the SATL complies (see SPC-4) (e.g., 06h for SPC-4).
NORMACA	The SATL shall set this bit to zero to indicate the SATL does not support the NACA bit in the CONTROL byte (see 6.5).
HISUP	Unspecified (see 3.4.2)
RESPONSE DATA FORMAT	The SATL shall set this field to 2h.
ADDITIONAL LENGTH	The SATL shall set this field to the length of the INQUIRY data that follows.
SCCS	Unspecified (see 3.4.2)
ACC	Unspecified (see 3.4.2)
TPGS	Unspecified (see 3.4.2)
3PC	Unspecified (see 3.4.2)
PROTECT	Unspecified (see 3.4.2)
ENCSERV	Unspecified (see 3.4.2)
MULTIP	Unspecified (see 3.4.2)
ADDR16	Unspecified (see 3.4.2)
WBUS16	Unspecified (see 3.4.2)

[a] If the INQUIRY command is sent to an incorrect logical unit then the SATL shall set the PERIPHERAL QUALIFIER field to 011b and shall set the PERIPHERAL DEVICE TYPE field to 1Fh.
[b] See 3.5.4.
[c] The full ATA IDENTIFY DEVICE data Model number field contents and the Firmware Revision field contents are returned in the ATA Information VPD page (see 12.4.2).
[d] The encoding used by the SPC-4 standard for INQUIRY version descriptors and the encoding used by the ATA8-ACS standard for ATA IDENTIFY DEVICE major and minor version numbers differ. The two standards may not define values for the same revisions.

Table 12 — Standard INQUIRY data fields (part 2 of 3)

Field	Description or reference
SYNC	Unspecified (see 3.4.2)
CMDQUE	Unspecified (see 3.4.2)
T10 VENDOR IDENTIFICATION	The SATL shall set the T10 VENDOR IDENTIFICATION field to 'ATA¬¬¬¬¬'[b].
PRODUCT IDENTIFICATION[c]	The SATL shall set the PRODUCT IDENTIFICATION field to a representation of the first 16 bytes of the ATA IDENTIFY DEVICE data Model number field, where each pair of bytes are swapped to create a valid ASCII string format: 1) byte 0 contains ATA IDENTIFY DEVICE word 27 bits 15:8 (i.e., byte 1); 2) byte 1 contains ATA IDENTIFY DEVICE word 27 bits 7:0 (i.e., byte 0); 3) byte 2 contains ATA IDENTIFY DEVICE word 28 bits 15:8 (i.e., byte 3); 4) byte 3 contains ATA IDENTIFY DEVICE word 28 bits 7:0 (i.e., byte 2); ... 15) byte 14 contains ATA IDENTIFY DEVICE word 34 bits 15:8 (i.e., byte 15); and 16) byte 15 contains ATA IDENTIFY DEVICE word 34 bits 7:0 (i.e., byte 14).
PRODUCT REVISION LEVEL[c]	The SATL shall set the PRODUCT REVISION LEVEL field to a four byte ASCII character representation of the ATA IDENTIFY DEVICE data Firmware revision field. Each pair of bytes are swapped to create a valid ASCII string format. Since the ATA IDENTIFY DEVICE data Firmware revision field contains eight ASCII characters and the standard INQUIRY data PRODUCT REVISION LEVEL field is four ASCII characters, the SATL shall select four of the eight ASCII characters from the ATA IDENTIFY DEVICE data Firmware revision field to return in the PRODUCT REVISION LEVEL field as follows: a) If the ATA IDENTIFY DEVICE data words 26:25 are set to four ASCII spaces (i.e., 2020_2020h), then the four ASCII characters selected shall contain: 1) byte 0 contains ATA IDENTIFY DEVICE data word 23 bits 15:8 (i.e., byte 1); 2) byte 1 contains ATA IDENTIFY DEVICE data word 23 bits 7:0 (i.e., byte 0); 3) byte 2 contains ATA IDENTIFY DEVICE data word 24 bits 15:8 (i.e., byte 3); and 4) byte 3 contains ATA IDENTIFY DEVICE data word 24 bits 7:0 (i.e., byte 2); or b) If the ATA IDENTIFY DEVICE data words 26:25 are not set to four ASCII spaces (i.e., 2020_2020h), then the four ASCII characters selected shall contain: 1) byte 0 contains ATA IDENTIFY DEVICE data word 25 bits 15:8 (i.e., byte 5); 2) byte 1 contains ATA IDENTIFY DEVICE data word 25 bits 7:0 (i.e., byte 4); 3) byte 2 contains ATA IDENTIFY DEVICE data word 26 bits 15:8 (i.e., byte 7); and 4) byte 3 contains ATA IDENTIFY DEVICE data word 26 bits 7:0 (i.e., byte 6).
CLOCKING	Unspecified (see 3.4.2)
QAS	Unspecified (see 3.4.2)
IUS	Unspecified (see 3.4.2)

[a] If the INQUIRY command is sent to an incorrect logical unit then the SATL shall set the PERIPHERAL QUALIFIER field to 011b and shall set the PERIPHERAL DEVICE TYPE field to 1Fh.

[b] See 3.5.4.

[c] The full ATA IDENTIFY DEVICE data Model number field contents and the Firmware Revision field contents are returned in the ATA Information VPD page (see 12.4.2).

[d] The encoding used by the SPC-4 standard for INQUIRY version descriptors and the encoding used by the ATA8-ACS standard for ATA IDENTIFY DEVICE major and minor version numbers differ. The two standards may not define values for the same revisions.

Table 12 — Standard INQUIRY data fields (part 3 of 3)

Field	Description or reference
VERSION DESCRIPTOR 1 to VERSION DESCRIPTOR 8	The SATL shall include version descriptors (see SPC-4) for: a) the SCSI Architecture Model standard (e.g., SAM-4); b) this standard; c) the SCSI Primary Commands standard (e.g., SPC-4); d) the SCSI Block Commands standard (e.g., SBC-3); e) if the SATL receives SCSI commands through a SCSI target port (see figure 5 in 5.1), the version of the transport protocol to which the SCSI target port was designed; f) if the SATL sends ATA commands through a SAS STP initiator port (see figure 7 in 5.1), the version of SAS (e.g., SAS-2) to which the SAS STP initiator port was designed; and g) the version(s) of ATA standards (e.g., ATA8-ACS and ATA8-AAM) to which the ATA device claims compliance in the ATA IDENTIFY DEVICE data word 80 (i.e., Major version number) and ATA IDENTIFY DEVICE data word 81 (i.e., Minor version number), and words 222 and 223.[d]
Vendor specific parameters	Unspecified (see 3.4.2)

[a] If the INQUIRY command is sent to an incorrect logical unit then the SATL shall set the PERIPHERAL QUALIFIER field to 011b and shall set the PERIPHERAL DEVICE TYPE field to 1Fh.
[b] See 3.5.4.
[c] The full ATA IDENTIFY DEVICE data Model number field contents and the Firmware Revision field contents are returned in the ATA Information VPD page (see 12.4.2).
[d] The encoding used by the SPC-4 standard for INQUIRY version descriptors and the encoding used by the ATA8-ACS standard for ATA IDENTIFY DEVICE major and minor version numbers differ. The two standards may not define values for the same revisions.

8.2 LOG SENSE command

8.2.1 LOG SENSE command overview

The LOG SENSE command provides a means for the application client to retrieve statistical or other operational information maintained by the SCSI target device about the SCSI target device or its logical units. Table 13 shows the translation for fields specified in the LOG SENSE CDB.

Table 13 — LOG SENSE CDB field translations

Field	Description or reference
OPERATION CODE	Set to 4Dh. The SATL shall implement support for this field by returning the log page data for the particular page requested.
PPC	Unspecified (see 3.4.2)
SP	Unspecified (see 3.4.2)
PC	8.2.2
PAGE CODE	8.2.3
SUBPAGE CODE	Unspecified (see 3.4.2)
PARAMETER POINTER	Unspecified (see 3.4.2)
ALLOCATION LENGTH	The SATL shall implement support for this field as defined in SPC-4.
CONTROL	6.5

8.2.2 PC (page control) field

The SATL interpretation and support of the page control values is shown in table 14.

Table 14 — PC field

Code	SAT Translation
00b	Unspecified (see 3.4.2)
01b	Supported
10b	Unspecified (see 3.4.2)
11b	Unspecified (see 3.4.2)

8.2.3 PAGE CODE and SUB PAGE CODE fields

The SATL shall support these field as defined in SPC-4. The SATL emulation for support of the PAGE CODE field is provided in table 15.

Table 15 — PAGE CODE / SUB PAGE CODE fields

Page Code	Subpage Code	Description
00h	00h	Supported Log Pages log page: The SATL shall implement this page by returning a list of supported log pages (see 10.2.3).
10h	00h	Self-Test Results log page: The SATL shall determine if the ATA SMART self-test is supported from the ATA IDENTIFY DEVICE data word 84 bit 1. If the ATA SMART self-test is not supported (i.e., word 84 bit 1 is set to zero) then the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the ATA SMART self-test is supported (i.e., word 84 bit 1 is set to one) then the SATL shall return the translated Self-Test Results log page to the application client (see 10.2.5).
2Fh	00h	Informational Exceptions log page: The SATL shall determine if the ATA SMART feature set is supported from the ATA IDENTIFY DEVICE data word 82 bit 0. If the ATA SMART feature set is not supported (i.e., word 82 bit 0 is set to zero) then the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the ATA SMART feature set is supported (i.e., word 82 bit 0 is set to one) then the SATL shall determine if the ATA SMART feature set is enabled or disabled from the ATA IDENTIFY DEVICE data word 85 bit 0. If the ATA SMART feature set is disabled (i.e., word 85 bit 0 is set to zero) then the SATL shall return a CHECK CONDITION status with the sense key set to ABORTED COMMAND and additional sense code set to ATA DEVICE FEATURE NOT ENABLED. If the ATA SMART feature set is enabled (i.e., word 85 bit 0 is set to one) then the SATL shall return the translated Informational Exceptions log page to the application client (see 10.2.6.1).
All others	All	Unspecified (see 3.4.2)

8.3 LOG SELECT command

8.3.1 LOG SELECT command overview

The LOG SELECT command provides a means for the application client to manage statistical information maintained by the SCSI target device about the SCSI device target or its logical units. Table 16 shows the translations of the fields specified in the LOG SELECT CDB.

Table 16 — LOG SELECT CDB field translations

Field	Description or Reference
OPERATION CODE	Set to 4Ch.
PCR	Unspecified (see 3.4.2)
SP	Unspecified (see 3.4.2)
PC	8.3.2
PAGE CODE	8.3.3
SUBPAGE CODE	8.3.3
PARAMETER LIST LENGTH	The SATL shall implement support for this field as defined in SPC-4.
CONTROL	6.5

8.3.2 Page control field translations

Table 17 shows the SATL interpretation of the PC field.

Table 17 — Page control field translation

Code	Translation
00b	Unspecified (see 3.4.2)
01b	Supported
10b	Unspecified (see 3.4.2)
11b	Unspecified (see 3.4.2)

8.3.3 PAGE CODE field and SUBPAGE CODE field translations

SATL emulation of the page code and subpage code is shown in table 18.

Table 18 — LOG SELECT PAGE CODE field and SUBPAGE CODE field translations

Page Code	Subpage Code	Description or reference
0Fh	00h	Application client log page. See 10.2.2 for the translation of the application client log page.
All others		Unspecified (see 3.4.2)

8.4 MODE SELECT (6) command

8.4.1 MODE SELECT (6) command overview

The MODE SELECT(6) command (see SPC-4) provides a means for an application client to specify medium, logical unit, or peripheral device parameters to a device server in the SATL. Device servers that implement the MODE SELECT (6) command shall also implement the MODE SENSE (6) command. Application clients

should send a MODE SENSE (6) command prior to each MODE SELECT (6) command to determine supported mode pages, page lengths, and other parameters.

The Mode Page Policy VPD page should be implemented (see 10.3.5). After a logical unit reset, the SATL shall set all mode page values to saved or default values. See 10.1 for supported mode pages.

8.4.2 MODE SELECT (6) CDB fields

The SATL shall support MODE SELECT (6) CDB fields as shown in table 19.

Table 19 — MODE SELECT (6) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 15h. The SATL shall modify logical unit, or peripheral device parameters for supported mode pages and parameters as specified in mode pages received from the application client. Some operational parameters in individual pages are provided via ATA (see 10.1).
SP	Unspecified (see 3.4.2)
PF	If this bit is set to zero (i.e., specifies that mode pages are vendor specific), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. The SATL shall support this bit being set to one (i.e., specifies that all mode page formats correspond to SPC-4 and SBC-3 mode page formats).
PARAMETER LIST LENGTH	Unspecified (see 3.4.2)
CONTROL	6.5

8.5 MODE SELECT (10) command

The MODE SELECT (10) command (see SPC-4) provides a means for an application client to set parameters in the device server in a SATL. It is a complementary command to the MODE SENSE(10) command and shall be implemented as in table 20.

Table 20 — MODE SELECT (10) CDB field translations

Field	Description or Reference
OPERATION CODE	Set to 55h.
SP	As specified for MODE SELECT (6), see 8.4
PF	As specified for MODE SELECT (6), see 8.4
PARAMETER LIST LENGTH	As specified for MODE SELECT (6), see 8.4
CONTROL	As specified for MODE SELECT (6), see 8.4

Device servers that implement the MODE SELECT (10) command shall also implement the MODE SENSE (10) command. See 10.1 for supported mode pages.

8.6 MODE SENSE (6) command

8.6.1 MODE SENSE (6) command overview

The MODE SENSE (6) command (see SPC-4) provides a means for a device server in a SATL to report parameters to an application client. It is a complementary command to the MODE SELECT(6) command.

Device servers that implement the MODE SENSE (6) command shall also implement the MODE SELECT(6) command. See 10.1 for supported mode pages.

8.6.2 MODE SENSE (6) CDB fields

The SATL shall support MODE SENSE (6) CDB fields as shown in table 21.

Table 21 — MODE SENSE (6) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 1Ah. The SATL shall return the requested mode pages to the application client. Some operational parameters in individual pages are gathered by issuing ATA commands (see 10.1).
DBD	A DBD bit set to zero specifies that zero or more block descriptors may be returned in MODE SENSE data. The SATL shall support only the mode parameter block descriptor format for direct-access block devices.
PC	Current values (i.e., the PC field is set to 00b) shall be supported. Reporting changeable, saveable, and default values is unspecified (see 3.4.2).
PAGE CODE	This field specifies the particular mode page requested (see 10.1). If the SATL does not support the specified mode page, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
SUB PAGE CODE	This field specifies the sub page code within the page code specified by PAGE CODE field that is requested by the application client (see 10.1).
ALLOCATION LENGTH	Unspecified (see 3.4.2)
CONTROL	6.5

8.7 MODE SENSE (10) command

The MODE SENSE (10) command (see SPC-4) provides a means for a device server in a SATL to report parameters to an application client. It is a complementary command to the MODE SELECT(10) command and shall be implemented as shown in table 22.

Table 22 — MODE SENSE (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 5Ah.
LLBA	Unspecified (see 3.4.2)
DBD	As defined for MODE SELECT (6), see 8.6
PC	As defined for MODE SELECT (6), see 8.6
PAGE CODE	As defined for MODE SELECT (6), see 8.6
SUB PAGE CODE	As defined for MODE SELECT (6), see 8.6
ALLOCATION LENGTH	As defined for MODE SELECT (6), see 8.6
CONTROL	As defined for MODE SELECT (6), see 8.6

SATLs that implement the MODE SENSE (10) command shall also implement the MODE SELECT(10) command. See 10.1 for supported mode pages.

8.8 READ BUFFER command

8.8.1 READ BUFFER command overview

The READ BUFFER command (see SPC-4) is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing memory in the SCSI device and the integrity of a service delivery subsystem. This command shall not alter the medium. Table 23 shows the translation for fields specified in the CDB for the READ BUFFER command.

Table 23 — READ BUFFER CDB field translations

Field	Description or reference
OPERATION CODE	Set to 3Ch. The SATL shall: a) send an ATA READ BUFFER command to the ATA device; or b) emulate the specified function (i.e., if supported); depending on the values for the BUFFER ID field and MODE field (see 8.8.2.1).
MODE	8.8.2
BUFFER ID	If the BUFFER ID field is set to 00h then the SATL shall return information describing or data read from the sector buffer in the ATA device, depending on the value in the MODE field (see 8.8.2). Otherwise, the translation is unspecified (see 3.4.2).
BUFFER OFFSET	The translation of this field depends on the contents of the MODE field (see 8.8.2).
ALLOCATION LENGTH	The translation of this field depends on the contents of the MODE field (see 8.8.2).
CONTROL	6.5

The logical sector buffer in a ATA device shall be used to emulate the READ BUFFER command, so the size of the buffer is limited to 512 bytes for data buffer and echo buffers.

8.8.2 MODE field

8.8.2.1 MODE field overview

Table 24 describes values of the MODE field that the SATL shall support.

Table 24 — MODE field

Code	Description or reference
02h (i.e., Data mode)	If BUFFER ID field is set to 00h, then the translation shall be to the ATA READ BUFFER command (see 8.8.2.2). Otherwise, the translation is unspecified (see 3.4.2).
03h (i.e., Descriptor mode)	8.8.2.3
All others	Unspecified (see 3.4.2)

8.8.2.2 Data mode

If the BUFFER ID field is set to 00h, the BUFFER OFFSET field is set to 00h, and the ALLOCATION LENGTH field is set to 512, then the SATL shall return 512 bytes of data.

If the BUFFER ID field is set to 00h, the BUFFER OFFSET field is set to 00h,and the ALLOCATION LENGTH field is set to a value other than 512, then the SATL shall either:

a) return the lesser of 512 bytes of data or the number of bytes specified in the ALLOCATION LENGTH field from the buffer in the ATA device by sending an ATA READ BUFFER command to the ATA device; or
b) terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

If the BUFFER ID field is set to 00h and the BUFFER OFFSET field is set to a value other than 00h then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

The SATL may support a value other than 00h in the BUFFER ID field. If the SATL supports a value other than 00h in the BUFFER ID field then the implementation shall be as defined in SPC-4.

A WRITE BUFFER command may be sent to the same buffer ID before it is read with the READ BUFFER command.

8.8.2.3 Descriptor mode

If the ALLOCATION LENGTH field is set to less than four, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the ALLOCATION LENGTH field is set to four or greater, then the SATL shall return four bytes of data describing the requested buffer, including the OFFSET BOUNDARY field and the BUFFER CAPACITY field.

If the BUFFER ID field is set to zero then the SATL shall return:

a) OFFSET BOUNDARY field set to 9h (i.e., 512 bytes); and
b) BUFFER CAPACITY field set to 200h (i.e., 512 bytes).

The SATL may support a value other than zero in the BUFFER ID field and the implementation is unspecified.

8.9 REQUEST SENSE command

8.9.1 REQUEST SENSE command overview

The REQUEST SENSE command requests any available sense data to be returned to the application client.

If the SCSI transport protocol for the SATL supports autosense (see 3.1.29), then the SATL shall return the sense data associated with a CHECK CONDITION status using autosense. Otherwise, the SATL shall return return contingent allegiance (see SAM-2) sense data in response to the REQUEST SENSE command.

The SATL shall determine if there is contingent allegiance (see SAM-2) sense data to return to the application client. To determine if there is power condition sense data to return, the SATL shall send the ATA CHECK POWER MODE command to the ATA device.

If the SATL has no sense data to return, then the SATL shall complete the REQUEST SENSE command with GOOD status with the sense key set to NO SENSE and the additional sense code set to NO ADDITIONAL SENSE DATA (see SPC-4). Table 25 lists examples of conditions where the SATL has sense data to return.

Table 25 — Special Request Sense behavior reference

Emulated device state	Reference
FORMAT UNIT in progress	8.9.2
SMART threshold exceeded condition	8.9.3
Stopped power condition	8.9.4
Unit attention condition established	8.9.5
Idle power condition	8.9.6
Standby power condition	8.9.7

Table 26 shows the fields in the REQUEST SENSE CDB.

Table 26 — REQUEST SENSE CDB field translations

Field	Description or reference
OPERATION CODE	Set to 03h. The SATL shall return any available sense data to the application client.
DESC	The SATL shall support this bit as defined in SPC-4 with the following exception: a) if the DESC bit is set to zero (i.e., requesting fixed format sense data), the SATL should return fixed format sense data for the ATA PASS-THROUGH commands.[a]
ALLOCATION LENGTH	Unspecified (see 3.4.2)
CONTROL	6.5

[a] SATLs compliant with previous versions of this standard return descriptor format sense data for the ATA PASS-THROUGH commands regardless of the value of the DESC bit.

8.9.2 Format operation in progress

If the SATL is processing a FORMAT UNIT command and the SATL receives a REQUEST SENSE command, then the SATL shall return GOOD status with the sense key set to NOT READY with the additional sense code set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS. The sense key specific bytes shall be set to progress indication as defined in SBC-3 and SPC-4.

8.9.3 SMART threshold exceeded condition

If:

a) the ATA device has the SMART feature set enabled (i.e., ATA IDENTIFY DEVICE data word 85 bit 0 is set to one);
b) the MRIE field in the Informational Exceptions Control mode page is set to 6h (see 10.1.8.2);
c) the DEXCPT bit in the Informational Exceptions Control mode page is set to zero; and
d) the most recent ATA SMART RETURN STATUS command to the ATA device indicates that the error threshold has been exceeded;

then the SATL shall:

a) return parameter data containing sense data with the sense key set to NO SENSE with the additional sense code set to HARDWARE IMPENDING FAILURE GENERAL HARD DRIVE FAILURE; and
b) complete the REQUEST SENSE command with GOOD status.

8.9.4 Stopped power condition

If the emulated logical unit is in the stopped power condition and there is no sense data to return for a previously returned CHECK CONDITION status, then the SATL shall:

1) return parameter data containing sense data with the sense key set to NO SENSE with the additional sense code set to NO ADDITIONAL SENSE DATA or LOGICAL UNIT NOT READY, INIITALIZING COMMAND REQUIRED; and
2) complete the REQUEST SENSE command with GOOD status.

Sense data returned for a previously returned CHECK CONDITION status resulting from a media access command or a TEST UNIT READY command received when the logical unit is in the stopped power condition is described in 8.13 (i.e., the TEST UNIT READY command) and 9.11 (i.e., the START STOP UNIT command).

8.9.5 Unit attention condition established

The SATL shall:

1) return parameter data containing sense data describing the unit attention condition (see SPC-4); and
2) complete the REQUEST SENSE command with GOOD status.

8.9.6 Idle power condition

If the emulated logical unit is in the idle power condition (e.g., after returning GOOD status to a START STOP UNIT command with the POWER CONDITION field set to IDLE), then the SATL shall return GOOD status with the sense key set to NO SENSE with the additional sense code set to:

a) POWER CONDITION CHANGE TO IDLE if the ATA CHECK POWER MODE command indicates Idle state; or
b) IDLE CONDITION ACTIVATED BY COMMAND if the logical unit entered the idle power condition due to a START STOP UNIT command or receipt of a command requiring the idle power condition.

8.9.7 Standby power condition

If the emulated logical unit is in the standby power condition (e.g., after returning GOOD status to a START STOP UNIT command with the POWER CONDITION field set to STANDBY), then the SATL shall return GOOD status with the sense key set to NO SENSE with the additional sense code set to:

a) POWER CONDITION CHANGE TO STANDBY if the ATA CHECK POWER MODE command indicates Standby state; or
b) STANDBY CONDITION ACTIVATED BY COMMAND if the logical unit entered the standby power condition due to a START STOP UNIT command or receipt of a command requiring the standby power condition.

8.10 SECURITY PROTOCOL IN command

The SECURITY PROTOCOL IN command provides a means for the application client to retrieve security information from a SCSI target device. Table 27 shows the translation for fields specified in the SECURITY PROTOCOL IN CDB.

Table 27 — SECURITY PROTOCOL IN CDB field translation

Field	Description or Reference
OPERATION CODE	Set to A2h. The SATL shall send the ATA TRUSTED RECEIVE command or ATA TRUSTED RECEIVE DMA command to the ATA device if the ALLOCATION LENGTH field is non-zero. Otherwise, the SATL shall send the ATA TRUSTED NON-DATA command to the ATA device.
SECURITY PROTOCOL	8.10.1
SECURITY PROTOCOL SPECIFIC	8.10.2
INC_512	8.10.3
ALLOCATION LENGTH	8.10.3
CONTROL	6.5

8.10.1 SECURITY PROTOCOL field

The SECURITY PROTOCOL field shall be copied to the ATA Security Protocol field.

8.10.2 SECURITY PROTOCOL SPECIFIC field

The SECURITY PROTOCOL SPECIFIC field shall be copied to the ATA SP Specific field.

8.10.3 ALLOCATION LENGTH field

If the ALLOCATION LENGTH field is zero, then the SATL shall use the ATA TRUSTED NON-DATA command with bit 24 of the LBA field set to one, instead of ATA TRUSTED RECEIVE command or ATA TRUSTED RECEIVE DMA command and the INC_512 bit shall be ignored.

If the INC_512 bit is set to one, then if the ALLOCATION LENGTH field contains a value greater than 0000_FFFFh, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. Otherwise, the ATA Transfer Length field shall be set to the contents of bits (15:0) of the ALLOCATION LENGTH field. After completion of the ATA TRUSTED RECEIVE command or ATA TRUSTED RECEIVE DMA command without error, the data shall be transferred to the SCSI application client.

If the INC_512 bit is set to zero, then if the ALLOCATION LENGTH field contains a value greater than 01FF_FE00h, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. Otherwise, the ATA Transfer Length field shall be translated from a number of bytes to a number of padded 512-byte units from the result of the following calculation:

ATA Transfer Length(15:0) = ((allocation length + 511) / 512)

After successful completion of the ATA TRUSTED RECEIVE command or ATA TRUSTED RECEIVE DMA command, the data shall be transferred to the SCSI application client up to the number of bytes specified in the ALLOCATION LENGTH field.

8.11 SECURITY PROTOCOL OUT command

The SECURITY PROTOCOL OUT command provides a means for the application client to send security information to a SCSI target device. Table 28 shows the translation for fields specified in the SECURITY PROTOCOL OUT CDB.

Table 28 — SECURITY PROTOCOL OUT CDB field translation

Field	Description or Reference
OPERATION CODE	Set to B5h. The SATL shall send the ATA TRUSTED SEND command or ATA TRUSTED SEND DMA command to the ATA device if the ALLOCATION LENGTH field is non-zero. Otherwise, the SATL shall send the ATA TRUSTED NON-DATA command to the ATA device.
SECURITY PROTOCOL	8.11.1
SECURITY PROTOCOL SPECIFIC	8.11.2
INC_512	8.11.3
TRANSFER LENGTH	8.11.3
CONTROL	6.5

8.11.1 SECURITY PROTOCOL field

The SECURITY PROTOCOL field shall be copied to the ATA Security Protocol field.

8.11.2 SECURITY PROTOCOL SPECIFIC field

The SECURITY PROTOCOL SPECIFIC field shall be copied to the ATA SP Specific field.

8.11.3 TRANSFER LENGTH field

If the TRANSFER LENGTH field is zero, then the SATL shall use the ATA TRUSTED NON-DATA command with bit 24 of the LBA field set to zero, instead of the ATA TRUSTED SEND command or ATA TRUSTED SEND DMA command and the INC_512 bit shall be ignored.

If the INC_512 bit is set to one, then if the TRANSFER LENGTH field contains a value greater than 0000_FFFFh, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. Otherwise, the ATA Transfer Length field shall be set to the contents of bits (15:0) of the ALLOCATION LENGTH field. The ATA TRUSTED SEND command or ATA TRUSTED SEND DMA command shall be used to transfer the data.

If the INC_512 bit is set to zero, then if the TRANSFER LENGTH field contains a value greater than 01FF_FE00h, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. Otherwise, the ATA Transfer Length field shall be translated from a number of bytes to a number of padded 512-byte units from the result of the following calculation:

ATA Transfer Length(15:0) = ((transfer length + 511) / 512)

The final data block may be padded (see SPC-4). The ATA TRUSTED SEND command or ATA TRUSTED SEND DMA command shall transfer the padded data for the number of blocks specified by the ATA Transfer Length field.

8.12 SEND DIAGNOSTIC command

8.12.1 SEND DIAGNOSTIC command overview

The SEND DIAGNOSTIC command provides a mechanism for an application client to request diagnostic operations to be performed on the target device, logical unit, or both. The SATL shall implement the default self-test feature (see SPC-4). Table 29 shows the translation for fields specified in the SEND DIAGNOSTIC CDB.

Table 29 — SEND DIAGNOSTIC CDB field translations

Field	Description or reference
OPERATION CODE	Set to 1Dh. See 8.12.2.
SELF-TEST CODE	8.12.2 and 8.12.3.
PF	Unspecified (see 3.4.2)
SELFTEST	8.12.3
DEVOFFL	If the DEVOFFL bit is set to zero, then the SATL shall process the command as specified in SPC-4. If the DEVOFFL bit is set to one, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
UNITOFFL	If the UNITOFFL bit is set to zero, then the SATL shall process the command as specified in SPC-4. If the UNITOFFL bit is set to one, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
PARAMETER LIST LENGTH	If the PARAMETER LIST LENGTH field is set to zero, then the SATL shall process the command as specified in SPC-4. If the PARAMETER LIST LENGTH field is not set to zero, then the SATL shall terminate the command with a CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.5

8.12.2 SELF-TEST CODE field

The SATL shall determine if the value in the SELF-TEST CODE field is valid depending on the value of the SELFTEST bit and the information that is reported by the ATA device with respect to the ATA SMART EXECUTE OFF-LINE IMMEDIATE command (see 8.12.3).

If the value of the SELF-TEST CODE field is valid, then the SATL shall process the command as described in table 30.

Table 30 — SELF-TEST CODE field translation (part 1 of 2)

Code	Name of test	Description of test
000b	Default self-test	Used when the SELFTEST bit is set to one.
001b	Background short self-test	The SATL shall perform the following: 1) return status for the SEND DIAGNOSTIC command as soon as the CDB has been validated and initialize the Self-Test Results log page (see 10.2.5 and SPC-4); and 2) send an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the ATA LBA Low field set to 1 (i.e., Execute SMART Short self-test routine immediately in off-line mode) to the ATA device.
010b	Background extended self-test	The SATL shall perform the following: 1) return status for the SEND DIAGNOSTIC command as soon as the CDB has been validated and initialize the Self-Test Results log page (see 10.2.5 and SPC-4); and 2) send an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the ATA LBA Low field set to 2 (i.e., Execute SMART Extended self-test routine immediately in off-line mode) to the ATA device.
011b	Reserved	
100b	Abort background self-test	If a previous SEND DIAGNOSTIC command specified a background self-test function and that self-test has not completed (see SPC-4), then the SATL shall send an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the ATA LBA Low field set to 127 (i.e., Abort off-line mode self-test routine) to the ATA device. If the ATA SMART EXECUTE OFF-LINE IMMEDIATE command completes without error, then the SATL shall return GOOD status. If the ATA command completes with an error, then the SATL shall respond as defined in SPC-4.
101b	Foreground short self-test	The SATL shall send an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the ATA LBA Low field set to 129 (i.e., Execute SMART Short self-test routine immediately in captive mode) to the ATA device. If the ATA SMART EXECUTE OFF-LINE IMMEDIATE command completes without error, then the SATL shall update the Self-Test Results log page prior to returning GOOD status. If the ATA command completes with an error, then the SATL shall first update the Self-Test Results log page (i.e., if supported, see SPC-4), and terminate the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to LOGICAL UNIT FAILED SELF-TEST.

Table 30 — SELF-TEST CODE field translation (part 2 of 2)

Code	Name of test	Description of test
110b	Foreground extended self-test	The SATL shall send an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the ATA LBA Low field set to 130 (i.e., Execute SMART Extended self-test routine immediately in captive mode) to the ATA device. If the ATA SMART EXECUTE OFF-LINE IMMEDIATE command completes without error, then the SATL shall update the Self-Test Results log page prior to returning GOOD status. If the ATA command completes with an error, then the SATL shall first update the Self-Test Results log page (i.e., if supported, see SPC-4), and then terminate the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and additional sense code set to LOGICAL UNIT FAILED SELF-TEST.
111b	Reserved	

8.12.3 SELFTEST bit

The SATL shall translate the SELFTEST bit according to whether or not the ATA device supports and has enabled the ATA SMART EXECUTE OFF-LINE IMMEDIATE command as shown in table 31.

Table 31 — SELFTEST bit

Code	ATA SMART EXECUTE OFF-LINE IMMEDIATE command[a]		SATL emulation
	supported	enabled	
0	no	n/a	The SATL shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
	yes	no	The SATL shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status with the sense key set to ABORTED COMMAND and the additional sense code set to ATA DEVICE FEATURE NOT ENABLED.
		yes	If the SELF-TEST CODE field is valid, then the SATL shall process the SEND DIAGNOSTIC command according to the value specified in the SELF-TEST CODE field as defined in 8.12.2.
1	no	n/a	The SATL shall send three ATA verify commands (see 3.1.24) to the ATA device with the ATA Count field set to one and the ATA LBA field set to: a) zero; b) the maximum user-addressable LBA; and c) an arbitrary number between zero and the maximum user-addressable LBA. If any of the three ATA verify commands ends with an error, then the SATL shall terminate the SEND DIAGNOSTIC command with a CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to LOGICAL UNIT FAILED SELF-TEST. If all three ATA verify commands complete without error[b], then the SATL shall return GOOD status.
	yes	no	
	yes		The SATL shall send an ATA SMART EXECUTE OFF-LINE IMMEDIATE command with the ATA LBA Low field set to 129 (i.e., Execute SMART Short self-test routine immediately in captive mode) to the ATA device. If the ATA EXECUTE OFF-LINE IMMEDIATE command completes without error, then the SATL shall return GOOD status. If the ATA EXECUTE OFF-LINE IMMEDIATE command completes with an error, then the SATL shall terminate the SEND DIAGNOSTIC command with a CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to LOGICAL UNIT FAILED SELF-TEST.

[a] The SATL shall determine if the ATA SMART EXECUTE OFF-LINE IMMEDIATE command is supported and enabled based on the ATA IDENTIFY DEVICE data word 84 bit 1, and word 85 bit 0 (see ATA8-ACS).

[b] The SATL may retry any of the three ATA Verify commands if an ATA Verify command fails on the first attempt, and the retried command may specify an alternate LBA. If the retried command completes without error, then the SATL may consider the ATA Verify command as having completed without error.

8.13 TEST UNIT READY command

8.13.1 TEST UNIT READY command overview

The TEST UNIT READY command is used to determine whether the device is ready (see table 32).

Table 32 — TEST UNIT READY CDB field translations

Field	Description or reference
OPERATION CODE	Set to 09h. See 8.13.2.
CONTROL	6.5

8.13.2 TEST UNIT READY command translation

The SATL processes the TEST UNIT READY command as follows:

1) If any condition exists that prevents the SATL from issuing commands to the ATA device, then the SATL should terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY with the additional sense code of LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE;
2) If the device is in the stopped power condition as the result of processing a START STOP UNIT command (see 9.11), then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code of LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED;
3) If the ATA device is processing a self-test in the foreground mode, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to NOT READY, and the additional sense code set to LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS;
4) If the SATL is processing a FORMAT UNIT command for the emulated device (see 9.2), then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS;
5) If the ATA device supports the Removable Media feature set (i.e., ATA IDENTIFY DEVICE data word 82 bit 2 is set to one), then the SATL shall send an ATA GET MEDIA STATUS command to the ATA device. If the ATA device completes the command with the NM bit set to one in the Error register, then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY and the additional sense code of MEDIUM NOT PRESENT; and
6) If the ATA device completed the most recent ATA command with the DF bit set to one in the Status register, then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code of LOGICAL UNIT FAILURE.

If none of the conditions defined in items 1 through 6 are valid, then the SATL shall send an ATA CHECK POWER MODE command to the ATA device, and:

a) If the ATA CHECK POWER MODE command completes with an error, then the SATL shall terminate the TEST UNIT READY command with CHECK CONDITION status with the sense key set to NOT READY, and the additional sense code set to LOGICAL UNIT DOES NOT RESPOND TO SELECTION; or
b) If the ATA CHECK POWER MODE command completes without error, then the SATL shall complete the TEST UNIT READY command with GOOD status.

8.14 WRITE BUFFER command

8.14.1 WRITE BUFFER command overview

The WRITE BUFFER command (see SPC-4) is used in conjunction with the READ BUFFER command as a diagnostic function for testing logical unit memory in the SCSI target device and the integrity of a service delivery subsystem. An additional mode is provided for downloading and saving microcode.

Table 33 shows the translation for fields specified in the WRITE BUFFER CDB.

Table 33 — WRITE BUFFER CDB field translations

Field	Description or reference
OPERATION CODE	Set to 3Bh. The SATL shall: a) send an ATA WRITE BUFFER command to the ATA device; b) send an ATA DOWNLOAD MICROCODE command to the ATA device; or c) emulate the specified function, depending on the values in the BUFFER ID field and MODE field (see 8.14.2.1).
MODE	8.14.2.1
BUFFER ID	If the BUFFER ID field is set to 00h then the SATL shall: a) transfer data to the buffer in the ATA device; b) download microcode to the ATA device; or c) emulate the specified WRITE BUFFER function, depending on the value set in the MODE field (see 8.14.2). If the BUFFER ID field is set to a value other than 00h then the translation is unspecified (see 3.4.2).
BUFFER OFFSET	The translation of this field depends on the contents of the MODE field (see 8.14.2.1).
PARAMETER LIST LENGTH	The translation of this field depends on the contents of the MODE field (see 8.14.2.1).
CONTROL	6.5

8.14.2 MODE field

8.14.2.1 MODE field overview

The MODE field specifies the function to be performed by the SATL.

Table 34 — MODE field

Code	Description or reference
02h (i.e., Write data)	Translated to the ATA WRITE BUFFER command (see 8.14.2.2).
05h (i.e., Download microcode and save)	Translated to the ATA DOWNLOAD MICROCODE command. The ATA Features field shall be set to 07h (i.e., indicating downloaded microcode is saved for immediate and future use) (see 8.14.2.3).
07h (i.e., Download microcode with offsets, save, and activate)	Translated to the ATA DOWNLOAD MICROCODE command. The ATA Features field shall be set to 03h (i.e., download microcode with offsets is saved for immediate and future use) (see 8.14.2.4).
All others	Unspecified (see 3.4.2)

8.14.2.2 Write data mode

In this mode, data transferred to the SATL from the application client is transmitted to the ATA device using the ATA WRITE BUFFER command.

If:

a) the BUFFER ID field is set to 00h;
b) the BUFFER OFFSET field is set to 00h; and
c) the PARAMETER LIST LENGTH field is set to 512,

then the SATL shall write the specified number of bytes to the buffer in the ATA device by sending an ATA WRITE BUFFER command to the ATA device.

If the BUFFER ID FIELD is set to 00h and either:

a) the BUFFER OFFSET field is set to a value other than 00h; or
b) the PARAMETER LIST LENGTH field is set to a value other than 512,

then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.

The SATL may support a value other than 00h in the BUFFER ID field. If the SATL supports a value other than 00h in the BUFFER ID field, then the implementation shall be as defined in SPC-4.

8.14.2.3 Download microcode mode 05h

In this mode, data transferred to the SATL from the application client is transmitted to the ATA device using the ATA DOWNLOAD MICROCODE command.

The SATL shall send an ATA DOWNLOAD MICROCODE command with the ATA Features field set to 07h to the ATA device when the SATL receives a WRITE BUFFER command with the MODE field set to 05h. The SATL shall check if the ATA DOWNLOAD MICROCODE command completed with an error (see ACS-2). If the ATA DOWNLOAD MICROCODE command completed with an error, then the SATL shall terminate the command with CHECK CONDITION status with the sense key and additional sense code set to values as described in clause 11. Otherwise, the SATL shall transfer the microcode image or control information from the application client to the ATA device, and then complete the WRITE BUFFER command with GOOD status.

After the ATA device reinitializes successfully running the new microcode image, the SATL shall establish a unit attention condition (see SAM-4) for the initiator port associated with all I_T nexuses except the I_T nexus on which the WRITE BUFFER command was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.

8.14.2.4 Download microcode mode 07h

In this mode, data transferred to the SATL from the application client is transmitted to the ATA device using the ATA DOWNLOAD MICROCODE command.

The SATL shall send an ATA DOWNLOAD MICROCODE command with the ATA field values specified in table 35 when the SATL receives a WRITE BUFFER command with the MODE field set to 07h. The SATL shall transfer the microcode or control information from the application client to the ATA device. The SATL shall check if the ATA DOWNLOAD MICROCODE command completed with an error. If the ATA DOWNLOAD MICROCODE command completed with an error, then the SATL shall terminate the command with CHECK CONDITION status with the sense key and additional sense code set as described in clause 11.

After the ATA device reinitializes successfully running the new microcode, the SATL shall establish a unit attention condition (see SAM-4) for the initiator port associated with all I_T nexuses except the I_T nexus on which the set of WRITE BUFFER commands was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.

Table 35 — Download Microcode Mode 07h ATA Field Values

ATA Field		Contents
Field Name	**Bits**	
Features	7:0	03h
LBA	27:24	Restricted
	23	0b
	22:8	BUFFER OFFSET field bits 23:9
	7	0b
	6:0	PARAMETER LIST LENGTH field bits 23:17
Count	7:0	PARAMETER LIST LENGTH field bits 16:9

If the PARAMETER LIST LENGTH field bits 8:0 is a non-zero value, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

If the BUFFER OFFSET field bits 8:0 is a non-zero value, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

If the PARAMETER LIST LENGTH field is greater than the contents of ATA IDENTIFY DEVICE data word 235 and the contents of ATA IDENTIFY DEVICE data word 235 is a non-zero value, then the SATL shall either translate the transfer into multiple ATA DOWNLOAD MICROCODE commands or terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

If the PARAMETER LIST LENGTH field is less than the contents of ATA IDENTIFY DEVICE data word 234 and the contents of ATA IDENTIFY DEVICE data word 234 is a non-zero value, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB

The SATL may translate a single WRITE BUFFER mode 07h request into multiple ATA DOWNLOAD MICROCODE commands.

If the combination of the BUFFER OFFSET field and PARAMETER LIST LENGTH field value result in a non-sequential or overlapping request and the ATA device returns an ATA abort status, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

9 SCSI Block Commands (SBC) command mapping

9.1 Translating LBA and transfer length and ATA command use constraints

9.1.1 Overview

A SATL may implement a direct logical block mapping of ATA logical sectors to SCSI logical blocks (see 9.1.2) or the SATL may implement indirect logical block mapping translation (see 9.1.3).

9.1.2 Direct logical block mapping model

If the SATL implements direct logical block mapping (see 3.1.35), then the logical block size indicated by the BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) shall equal the ATA logical sector size (see 3.1.16). The ATA LBA of an ATA logical sector shall equal the logical block address of the corresponding SCSI logical block.

9.1.3 Indirect logical block mapping model

If the SATL implements indirect block mapping (see 3.1.40), then the constraints of the direct logical block mapping model do not apply. The logical block size indicated by the BLOCK LENGTH IN BYTES field in the READ CAPACITY data (see 9.8.2 and 9.9.2) may not equal the ATA logical sector size (see 3.1.16) (e.g., SCSI logical block size of 520 bytes with an ATA Logical Sector Size of 512 bytes). The SATL translates between the SCSI LOGICAL BLOCK ADDRESS field and the ATA LBA in a vendor-specific manner. The result of a logical block address translated in one direction and then translated in the reverse direction shall yield the original logical block address.

9.1.4 Selection of ATA block commands

The ATA commands the SATL may use to implement the functions specified by SCSI block commands depend upon:

a) the value of the LOGICAL BLOCK ADDRESS field and TRANSFER LENGTH field specified in the SCSI CDB; and
b) the capabilities of the ATA device and the ATA host within the SATL.

Table 36 relates selection conditions to allowable ATA commands used to implement SCSI block storage data transfer commands. ATA commands listed in the Allowed ATA commands column shall not be used in the emulation of a SCSI block command if the prerequisite conditions listed in Selection Prerequisites columns are not met (i.e., the word 'yes' in a Selection Prerequisites column means the prerequisite shall be met before the SATL may use an ATA command listed in that row, and the word 'no' indicates the prerequisite need not be

met for the SATL to use the ATA command listed).

Table 36 — ATA commands used for SCSI block command translations

Selection Prerequisites [a]					Allowed ATA commands
Highest ATA logical sector accessed	ATA feature sets required to be supported and enabled [d]				
Required that the logical sector address is < 2^{28} [b]	48-bit Address [b]	ATA Features [c]	TCQ	NCQ	
no	no	no	no	no	FLUSH CACHE WRITE UNCORRECTABLE EXT
yes [e]	no	no	no	no	READ MULTIPLE READ SECTOR(S) READ VERIFY SECTOR(S) WRITE MULTIPLE WRITE SECTOR(S)
yes [e]	no	yes	no	no	READ DMA WRITE DMA
yes [e]	no	yes	yes	no	READ DMA QUEUED WRITE DMA QUEUED
no	yes	yes	no	no	FLUSH CACHE EXT READ DMA EXT WRITE DMA EXT WRITE DMA FUA EXT
no	yes	yes	yes	n/a	READ DMA QUEUED EXT WRITE DMA QUEUED EXT WRITE DMA QUEUED FUA EXT
no	yes	no	no	no	READ MULTIPLE EXT READ SECTOR(S) EXT READ VERIFY SECTOR(S) EXT WRITE MULTIPLE EXT WRITE MULTIPLE FUA EXT WRITE SECTOR(S) EXT
no	no	no	no	yes	READ FPDMA QUEUED WRITE FPDMA QUEUED

[a] If the SATL implements the direct mapping model (see 9.1.2) between ATA logical sectors and SCSI logical blocks, then this represents the last logical block transferred. If the SATL implements the indirect logical block mapping model, then this constraint is vendor-specific.

[b] If the ATA device supports neither the 48-bit Address feature set (i.e., ATA IDENTIFY DEVICE data word 83 bit 10 is set to zero) nor NCQ (see SATA-2.6) and the LBA of the logical sector is greater than (2^{28}-1), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

[c] The DMA prerequisite requires both the ATA host in the SATL and the ATA device to have the same DMA transfer mode enabled (i.e., ATA IDENTIFY DEVICE data word 49 bit 8 is set to one and at least one DMA mode is enabled in the ATA IDENTIFY DEVICE data word 63 or word 88).

[d] See ATA8-ACS.

[e] The SATL may transfer the number of logical blocks requested in the TRANSFER LENGTH field of the SCSI CDB by sending multiple ATA commands, each time incrementing the ATA LBA by the ATA Sector Count transferred.

The SATL may use the ATA commands listed in table 36 in the translation of SCSI read commands (see 3.1.77), SCSI write commands (see 3.1.81), SCSI write and verify commands (see 3.1.82), SCSI verify commands (see 3.1.80), and SCSI synchronize cache commands (see 3.1.78) if the prerequisites defined for the command as shown in table 36 are satisfied. The translations for specific SCSI block commands in clause 9 further constrain the use of the available ATA commands in implementing the translation.

The SATL emulation of the READ (6) command and the WRITE (6) command in which the TRANSFER LENGTH field is set to zero, shall translate the transfer length to 256, and send ATA commands that operate on the ATA logical sectors corresponding to the specified 256 SCSI logical blocks.

In all other cases, the SATL shall transfer or operate on the ATA logical sectors corresponding to the number of logical blocks specified by the SCSI command.

9.2 FORMAT UNIT command

9.2.1 FORMAT UNIT command overview

The FORMAT UNIT command verifies that all logical block addresses accessible to SCSI application clients are formatted and ready for data transfers. Table 37 shows the translation for fields in the FORMAT UNIT CDB.

Table 37 — FORMAT UNIT CDB field translations

Field	Description or reference
OPERATION CODE	Set to 04h. If no defect list header is provided or a defect list header is provided with the DCRT bit set to one, then the SATL shall return GOOD status without issuing any commands to the ATA device. If the SATL supports certification of media and a defect list header is provided with the DCRT bit set to zero, then the SATL shall certify the media as described in 9.2.5
FMTPINFO	Unspecified (see 3.4.2)
LONGLIST	Unspecified (see 3.4.2)
FMTDATA	If set to zero then no data shall be transferred from the application client. If set to one then the FORMAT UNIT parameter list shall be transferred from the application client. The SATL may accept a FORMAT UNIT parameter list (see 9.2.2). The SATL shall ignore any defect list descriptors and any other fields provided in the FORMAT UNIT parameter list.
DEFECT LIST FORMAT	If the DEFECT LIST FORMAT field is the mandatory format (000b) or the vendor specific format (110b), then the defect list length shall be zero (see SBC-3). If the DEFECT LIST FORMAT field is any other value, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CMPLIST	If a complete list is specified (i.e., the FMTDATA bit is set to one, and the CMPLIST bit is set to one), then the SATL shall terminate the command with a CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.
RTO_REQ	If the RTO_REQ bit is set to one and the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall terminate the command with CHECK CONDITION status with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB. If the SATL implements indirect logical block mapping, then the translation for this field is unspecified (see 3.4.2).
CONTROL	6.5

The SATL shall process commands received during the processing of the FORMAT UNIT command as specified in SBC-3.

9.2.2 FORMAT UNIT parameter list

If the FORMAT command CDB specifies a FMTDATA bit set to one, then the SATL shall accept a FORMAT UNIT parameter list consisting of a short or long defect list header and may accept an initialization pattern descriptor. The SATL shall ignore any defect descriptors provided. Table 38 defines the SATL handling of fields in the FORMAT UNIT defect list header.

Table 38 — SATL defect list header

Field	Description or reference
FOV	9.2.3 and 9.2.4
DPRY	The SATL shall ignore this field.
DCRT	9.2.3 and 9.2.5
STPF	Unspecified (see 3.4.2)
IP	9.2.3 and 9.2.6
IMMED	9.2.3
DEFECT LIST LENGTH	The SATL shall ignore any defect descriptors provided.

9.2.3 SATL defect list header field combinations

Table 39 describes the actions the SATL takes depending on the values set in the IMMED bit, the FOV bit, the DCRT bit, and the IP bit.

Table 39 — SATL defect list header field combinations

<table>
<tr><th>IMMED</th><th>FOV</th><th>DCRT</th><th>IP</th><th>Description of SATL processing</th></tr>
<tr><td>1</td><td>n/a</td><td>n/a</td><td>n/a</td><td rowspan="3">The SATL may complete the FORMAT UNIT command immediately with GOOD status.</td></tr>
<tr><td>n/a</td><td>0</td><td>n/a</td><td>n/a</td></tr>
<tr><td>n/a</td><td>1</td><td>1</td><td>0</td></tr>
<tr><td rowspan="3">0</td><td rowspan="3">1</td><td>0</td><td>0</td><td rowspan="3">If the SATL does not support media certification, then the SATL may terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD PARAMETER LIST. Otherwise, the SATL shall send the required ATA read commands and ATA write commands to certify and initialize the media as specified by DCRT bit and IP bit, and shall then return GOOD status if no unrecoverable read errors occur.</td></tr>
<tr><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>

9.2.4 FOV bit

The FOV bit may be set to one to include an initialization pattern descriptor and no defect descriptors, otherwise the SATL may terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD PARAMETER LIST.

9.2.5 DCRT bit

If the DCRT bit is set to zero and media certification is supported by the SATL, then the SATL shall send ATA verify commands (see 3.1.24) to access all the logical sectors on the medium of the ATA device that the SATL uses to emulate logical blocks accessible by the application client. For every unrecoverable read error that is encountered, the SATL shall send an ATA write command (see 3.1.26) to the defective logical sector to attempt to cause logical sector reallocation. The data written shall be the data pattern specified by the

initialization pattern descriptor, if any, or vendor-specific. After writing the affected logical sector, the SATL shall again send an ATA verify command to the same logical sector to verify the alternate logical sector is not defective. The process (e.g., verify, write, verify, write, etc.) shall repeat until the logical sector is verified successfully or the disk returns a fatal error other than an unrecoverable read error (e.g., device fault). See 5.4 for a description of error handling for multiple ATA command sequences.

9.2.6 IP bit

If the SATL supports an IP bit value of one and the IP bit is set to one, then the SATL shall process the command as follows:

a) If the ATA device supports the SCT LBA Write Same (see ACS-2) command and the value of the INITIALIZATION PATTERN LENGTH field in the initialization pattern descriptor is four, and the value of the IP MODIFIER field in the initialization pattern descriptor is zero, then the SATL should send an ATA SCT LBA Segment Access command to the ATA device with the Function Code field set to 0001b (i.,e., Repeat Write Pattern), with the Start field and the Count field set to initialize the area of the media accessible by the application client, and with the Pattern field set to the value of the INITIALIZATION PATTERN field from the FORMAT command initialization pattern descriptor; and
b) if the ATA SCT LBA Segment Access command is not used to write the initialization pattern, then the SATL shall write the specified pattern by issuing ATA write commands (see 3.1.26 and 9.1) to the ATA device.

If the IP bit is set to zero, then the SATL shall return GOOD status.

9.3 READ commands overview

9.3.1 READ commands operation code translation

This subclause applies to the translation of READ(6) command, READ(10) command, READ(12) command, and READ(16) command.

The SATL shall send ATA read commands (see 3.1.21) in accordance with the constraints specified in 9.1 to cause the ATA device to transfer the logical blocks specified in the SCSI read command (see 3.1.77).

If the SATL returns a CHECK CONDITION status with a sense key set to a value other than ILLEGAL REQUEST while processing the command, then the SATL may transfer a vendor-specific amount of data before terminating the command.

9.3.2 READ commands with FUA

If the SATL does not support FUA and the FUA bit is set to one, then the SATL shall terminate the READ (10) command, READ (12) command, or READ (16) command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The SATL shall process a SCSI read command with the FUA bit set to one as follows:

a) If the ATA device supports NCQ (i.e., ATA IDENTIFY DEVICE data word 76 bit 8 is set to one), then the SATL shall send an ATA READ FPDMA QUEUED command (see SATA-2.6) with the ATA FUA bit in the ATA Device field set to one;
b) If the ATA device supports TCQ (see ATA8-ACS) and there are outstanding ATA queued commands, then the SATL shall:
    1) wait until all ATA queued commands have completed;
    2) if the write cache is enabled (see ATA8-ACS) on the ATA device, then send an ATA verify command (see 3.1.24); and,
    3) send an ATA read command as specified in 9.3.1;
    or
c) If the ATA device supports neither NCQ nor TCQ or there are no outstanding ATA queued commands, then the SATL shall:
    1) if the write cache is enabled on the ATA device (see ATA8-ACS), send an ATA verify command (see 3.1.24); and
    2) send an ATA read command as specified in 9.3.1.

9.4 READ (6) command

The READ (6) command is used to request the device to transfer logical blocks of user data to the application client (see SBC-3). Table 40 shows the translation for fields in the READ (6) CDB.

Table 40 — READ (6) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 08h. See 9.3.1
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.21) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
TRANSFER LENGTH [a]	The transfer length shall be used to set the ATA Sector Count (see 3.1.22), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA Sector Count in the ATA read command (see 3.1.21) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.5

[a] A transfer length of zero specifies to transfer 256 logical blocks to the application client (see SBC-3).

9.5 READ (10) command

The SATL shall process the READ (10) command the same as the READ (6) command (see 9.3.1). with the additional fields in the CDB implemented as described in the table 41 and 9.3.2.

Table 41 — READ (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 28h. See 9.3.1 and 9.3.2.
RDPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.3.2
FUA_NV	The SATL may ignore the FUA_NV bit or the SATL may implement the FUA_NV bit as defined in SBC-3.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA read command (see 3.1.21) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH[a]	The transfer length is used to set the ATA Sector Count (see 3.1.22), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA Sector Count in the ATA read command (see 3.1.21) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall send as many ATA read commands as needed to satisfy the transfer length.
CONTROL	6.5

[a] A transfer length of zero specifies that a data transfer shall not take place.

9.6 READ (12) command

The SATL shall process the READ (12) command the same as the READ (10) command (see 9.3.1), with the fields in the CDB implemented as described in table 42 and 9.3.2.

Table 42 — READ (12) CDB field translations

Field	Description or reference
OPERATION CODE	Set to A8h. See 9.3.1 and 9.3.2.
RDPROTECT	As defined in READ (10) (see 9.5).
DPO	As defined in READ (10) (see 9.5).
FUA	As defined in READ (10) (see 9.5).
FUA_NV	As defined in READ (10) (see 9.5).
LOGICAL BLOCK ADDRESS	As defined in READ (10) (see 9.5).
GROUP NUMBER	As defined in READ (10) (see 9.5).
TRANSFER LENGTH[a]	As defined in READ (10) (see 9.5).
CONTROL	6.5
[a] A transfer length of zero specifies that a data transfer shall not take place.	

9.7 READ (16) command

The SATL shall process the READ (16) command the same as the READ (10) command (see 9.3.1), with the fields in the CDB implemented as described in table 43 and 9.3.2.

Table 43 — READ (16) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 88h. See 9.3.1 and 9.3.2.
RDPROTECT	As defined in READ (10) (see 9.5).
DPO	As defined in READ (10) (see 9.5).
FUA	As defined in READ (10) (see 9.5).
FUA_NV	As defined in READ (10) (see 9.5).
LOGICAL BLOCK ADDRESS	As defined in READ (10) (see 9.5).
GROUP NUMBER	As defined in READ (10) (see 9.5).
TRANSFER LENGTH[a]	As defined in READ (10) (see 9.5).
CONTROL	6.5
[a] A transfer length of zero specifies that a data transfer shall not take place.	

9.8 READ CAPACITY (10) command

9.8.1 READ CAPACITY (10) command overview

The READ CAPACITY (10) command (see SBC-3) requests that the device server transfer eight bytes of parameter data describing the capacity and medium format of the direct-access block device to the application client. Table 44 shows the translation for fields in the READ CAPACITY (10) CDB.

Table 44 — READ CAPACITY (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 25h. The SATL shall use ATA IDENTIFY DEVICE data to compute the ATA device's maximum user addressable medium capacity of the ATA device.
LOGICAL BLOCK ADDRESS	If the LOGICAL BLOCK ADDRESS field is not set to zero then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
PMI	If the PMI bit is not set to zero then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.5

9.8.2 READ CAPACITY (10) parameter data

The SATL shall return READ CAPACITY (10) parameter data as defined by SBC-3. Table 45 describes the translation of fields in the READ CAPACITY (10) parameter data.

Table 45 — READ CAPACITY (10) parameter data

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS[a]	If the SATL implements direct logical block mapping (see 3.1.35), then this field shall contain the lower of: a) the ATA maximum LBA (see 3.1.17); or b) FFFF_FFFFh. If the SATL implements indirect logical block mapping, then this field is unspecified (see 3.4.2).
LOGICAL BLOCK LENGTH IN BYTES[a]	If the SATL implements direct logical block mapping (see 3.1.35) then this field shall contain the ATA logical sector size (see 3.1.16). Otherwise this field is unspecified (see 3.4.2).

[a] The values reported in the RETURNED LOGICAL BLOCK ADDRESS field and the LOGICAL BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.51) is less than or equal to the ATA device capacity (see 3.1.10).

9.9 READ CAPACITY (16) command

9.9.1 READ CAPACITY (16) command overview

The READ CAPACITY (16) command (see SBC-3) requests that the device server transfer parameter data describing the capacity and medium format of the direct-access block device to the application client. Table 46 shows the translation for fields in the READ CAPACITY (16) CDB.

Table 46 — READ CAPACITY(16) CDB field translations

Field	Description or reference
OPERATION CODE / SERVICE ACTION	Set to 9Eh/10h.
LOGICAL BLOCK ADDRESS	As defined in READ CAPACITY (10) (see 9.8).
ALLOCATION LENGTH	Unspecified (see 3.4.2)
PMI	As defined in READ CAPACITY (10) (see 9.8).
CONTROL	6.5

9.9.2 READ CAPACITY (16) parameter data

The SATL shall return READ CAPACITY (16) parameter data as defined by SBC-3. Table 47 describes the translation of fields in the READ CAPACITY (16) parameter data.

Table 47 — READ CAPACITY (16) parameter data

Field	Description or reference
RETURNED LOGICAL BLOCK ADDRESS[a]	If the SATL implements direct logical block mapping (see 3.1.35), then this field shall contain the ATA maximum LBA (see 3.1.17). If the SATL implements indirect logical block mapping, then this field is unspecified (see 3.4.2).
LOGICAL BLOCK LENGTH IN BYTES[a]	As defined in READ CAPACITY (10) (see 9.8).
PROT_EN	Unspecified (see 3.4.2)
P_TYPE	Unspecified (see 3.4.2)
LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT	If the SATL implements direct logical block mapping (see 3.1.35) then this field shall contain the ATA logical sectors per physical sector exponent (see 5.7). If the SATL implements indirect logical block mapping (see 3.1.40), then this field is unspecified (see 3.4.2).
LOWEST ALIGNED LOGICAL BLOCK ADDRESS	If the SATL implements direct logical block mapping and the ATA logical sector alignment is zero, then this field shall be set to zero. Otherwise, this field shall contain the ATA logical sector alignment subtracted from the ATA logical sectors per physical sector (see 5.7). If the SATL implements indirect logical block mapping, then this field is unspecified.

[a] The values reported in the RETURNED LOGICAL BLOCK ADDRESS field and the BLOCK LENGTH IN BYTES field shall be such that the logical unit capacity (see 3.1.51) is less than or equal to the ATA device capacity (see 3.1.10).

9.10 REASSIGN BLOCKS command

9.10.1 REASSIGN BLOCKS command overview

The REASSIGN BLOCKS command requests that the SATL reassign defective logical blocks (see SBC-3). ATA devices do not support or have a direct translation for the REASSIGN BLOCKS command. Table 48 shows the translation for fields in the REASSIGN BLOCKS CDB.

Table 48 — REASSIGN BLOCKS CDB field translations

Field	Description or reference
OPERATION CODE	Set to 07h. See 9.10.2.
LONGLBA	See SBC-3
LONGLIST	See SBC-3
CONTROL	6.5

The REASSIGN BLOCKS command parameter list transferred from the application client contains the LBAs of logical blocks to be reassigned.

If the SATL implements direct logical block mapping (see 9.1.2), then the values set by the SATL in the ATA LBA of the ATA verify command(s) and ATA write command(s) shall equal the value(s) of the LBAs in the parameter list. Otherwise, the mapping is unspecified (see 3.4.2).

The SATL shall support the LONGLBA bit and the LONGLIST bit (see SBC-3).

9.10.2 REASSIGN BLOCKS operation code

The SATL shall accept a parameter list specifying LBAs of logical blocks to be reassigned (see SBC-3).

The SATL shall process each ATA LBA corresponding to LBAs specified in the parameter list as shown in figure 11.

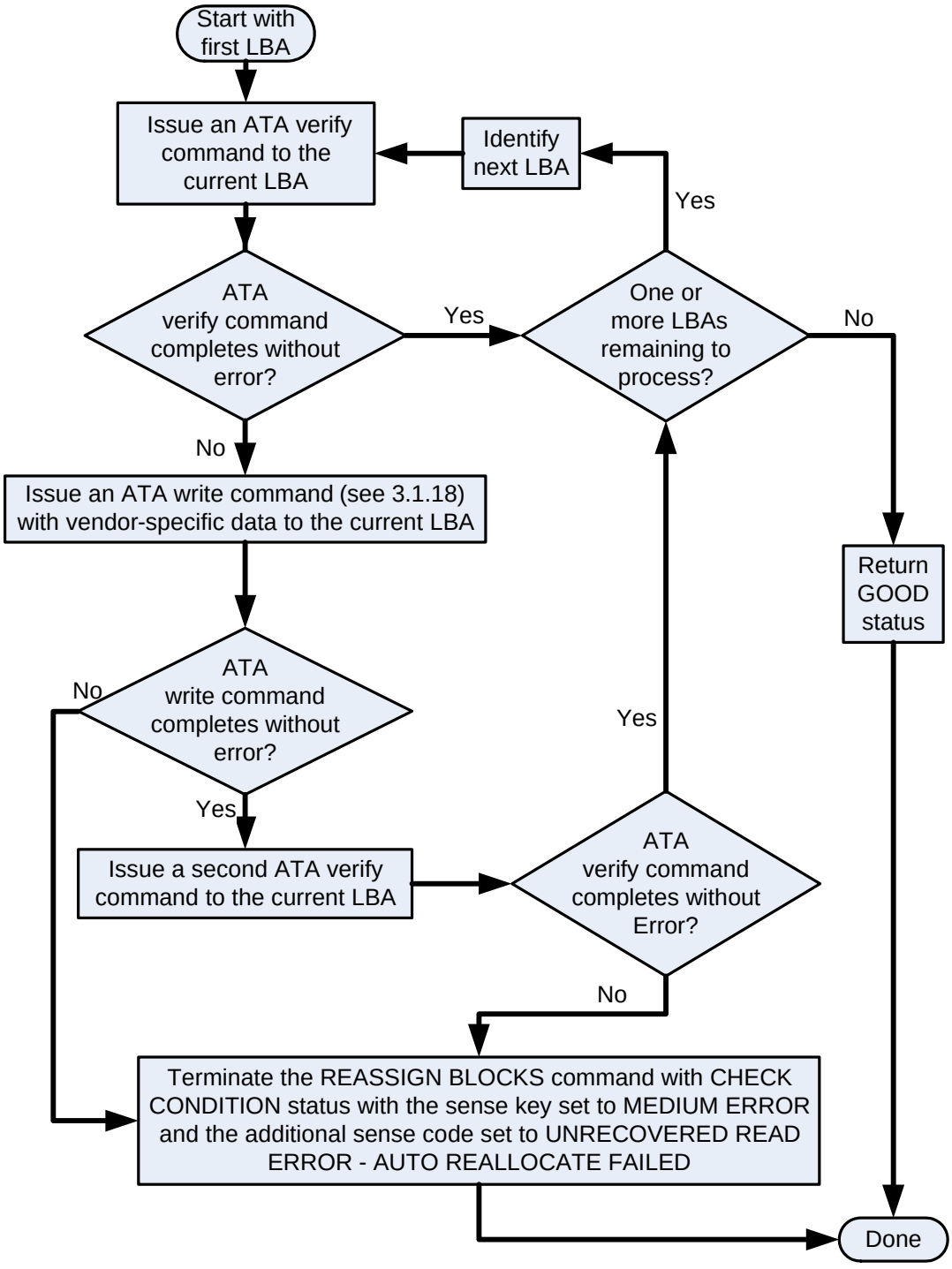


Figure 11 — REASSIGN BLOCKS command translation

9.11 START STOP UNIT command

9.11.1 START STOP UNIT command overview

The START STOP UNIT command provides a method for controlling the power state of a logical unit.

If a SATL receives a command that requires medium access while the device is in the Stopped state (see SBC-3), then the SATL shall return CHECK CONDITION status, with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED.

The POWER CONDITION field is used to specify that the logical unit be placed into a specific power condition or to adjust a timer as defined in table 49. If the POWER CONDITION field contains a value other than 0h, then the SATL shall not consider the ATA device to be in the stopped state (see 8.13.2). If this field is not supported and is set to a value other than 0h, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

Table 49 shows the translation for fields in the START STOP UNIT CDB.

Table 49 — START/STOP UNIT CDB field translations

Field	Description or reference
OPERATION CODE	Set to 1Bh. See 9.11.2 and 9.11.3.
IMMED	The SATL shall implement this field as defined in 9.11.2 and 9.11.3.
POWER CONDITION MODIFIER	See table 50 with the POWER CONDITION field set to 02h.
POWER CONDITION	See table 50.
NO_FLUSH	See 9.11.4
LOEJ	The SATL shall implement this field as defined in 9.11.3.
START	The SATL shall implement this field as defined in 9.11.3.
CONTROL	6.5

Table 50 describes the translations for the POWER CONDITION field of the START STOP UNIT CDB.

Table 50 — POWER CONDITION field translation (part 1 of 5)

Code	Name	Description or Reference
00h	START_VALID	The SATL shall process the LOEJ and START fields as defined in 9.11.3.
01h	ACTIVE	The SATL shall: 1) If the IMMED bit is set to one, then return GOOD status; 2) Send an ATA IDLE command to the ATA device with the ATA Feature field set to zero, the ATA Count field set to zero, and the ATA LBA field set to zero; 3) If the ATA IDLE command completes in step 2 with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 4) Send an ATA verify [a] command (see 3.1.24) to the ATA device with the ATA Sector Count set to one and the LBA set to a value between zero and the maximum LBA supported by the ATA device in its current configuration; 5) If the ATA verify command in step 4 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 6) If the ATA verify command in step 4 completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2) and the SATL shall no longer consider the ATA device to be in the stopped power state.

[a] For ATA devices compliant with previous versions of ATA, ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode after completion of the ATA verify command.

Table 50 — POWER CONDITION field translation (part 2 of 5)

Code	Name	Description or Reference
02h	IDLE	The SATL shall: 1) If the IMMED bit is set to one, then return GOOD status; 2) If the NO_FLUSH bit is set to zero, then send an ATA flush command (see 3.1.12) to the ATA device; 3) If the ATA flush command in step 2 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 4) If the ATA flush command in step 2 was sent and completes without error, then: A) If the POWER CONDITION MODIFIER field is set to zero, then: 1) Send an ATA IDLE command to the ATA device with the ATA Feature field set to zero, the ATA Count field set to zero, and the ATA LBA field set to zero; 2) If the ATA IDLE command in step A-1 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 3) If the ATA IDLE command in step A-1 completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2); and B) If the POWER CONDITION MODIFIER field is not set to zero, then: 1) Send an ATA IDLE command to the ATA device with the ATA Feature field set to zero, the ATA Count field set to zero, and the ATA LBA field set to zero; 2) If the ATA IDLE command in step B-1 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 3) If the ATA IDLE command in step B-1 completes without error and ATA IDENTIFY DEVICE data word 84, bit 13 is set to zero and the IMMED bit is set to zero, then return GOOD status (see 9.11.2); 4) If the ATA IDLE command in step B-1 completes without error and ATA IDENTIFY DEVICE data word 84, bit 13 is set to one, then send an ATA IDLE IMMEDIATE command to the ATA device with the ATA Feature field set to 44h, the ATA Count field set to zero, and the ATA LBA field set to 55_4E4Ch; 5) If the ATA IDLE IMMEDIATE command in step B-4 completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 6) If the ATA IDLE IMMEDIATE command in step B-4 completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2).

[a] For ATA devices compliant with previous versions of ATA, ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode after completion of the ATA verify command.

Table 50 — POWER CONDITION field translation (part 3 of 5)

Code	Name	Description or Reference
03h	STANDBY	The SATL shall: 1) If the IMMED bit is set to one, then return GOOD status; 2) If the NO_FLUSH bit is set to zero, then send an ATA flush command (see 3.1.11) to the ATA device; 3) If the ATA flush command in step 2 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 4) If the ATA flush command in step 2 completes without error, then the SATL shall send an ATA STANDBY command to the ATA device with the ATA Count field set to zero; 5) If the ATA STANDBY command in step 4 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 6) If the ATA STANDBY command in step 4 completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2).

[a] For ATA devices compliant with previous versions of ATA, ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode after completion of the ATA verify command.

Table 50 — POWER CONDITION field translation (part 4 of 5)

Code	Name	Description or Reference
07h	LU_CONTROL	The SATL shall: 1) if the IMMED bit is set to one, then return GOOD status; 2) send an ATA CHECK POWER MODE command to the ATA device; 3) If the ATA CHECK POWER MODE command in step 2 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 4) if the ATA Count field returned from the ATA CHECK POWER MODE command is 00h, then: 1) send an ATA STANDBY command to the ATA device with the ATA Count field set to the previously saved value of the ATA standby timer (see 10.1.9.3); 2) if the ATA STANDBY command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 3) if the ATA STANDBY command completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2); 5) if the ATA Count field returned from the ATA CHECK POWER MODE command is 80h, then: 1) send an ATA IDLE command to the ATA device with the ATA Count field set to the previously saved value of the ATA standby timer (see 10.1.9.3); 2) if the ATA IDLE command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 3) if the ATA IDLE command completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2); and

[a] For ATA devices compliant with previous versions of ATA, ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode after completion of the ATA verify command.

Table 50 — POWER CONDITION field translation (part 5 of 5)

Code	Name	Description or Reference
07h (con't)	LU_CONTROL (con't)	6) if the ATA Count field returned from the ATA CHECK POWER MODE command is 40h, 41h or FFh, then: 1) send an ATA IDLE command to the ATA device with the ATA Count field set to the previously saved value of the ATA standby timer (see 10.1.9.3); 2) if the ATA IDLE command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 3) if the ATA IDLE command completes without error, then send an ATA verify command (see 3.1.24) to the ATA device with the ATA Sector Count set to one and the LBA set to a value between zero and the maximum LBA supported by the ATA device in its current configuration; 4) if the ATA verify command completes with any error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 5) if the ATA verify command completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2).
0Bh	FORCE_STANDBY_0	If ATA IDENTIFY DEVICE data word 49, bit 13 is set to one and the standby timer is enabled, then the SATL shall: 1) if the IMMED bit is set to one, then return GOOD status; 2) if the NO_FLUSH bit is set to zero, then send an ATA flush command (see 3.1.11) to the ATA device; 3) if the ATA flush command in step 2 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 4) if the ATA flush command in step 2 completes without error, then send an ATA STANDBY IMMEDIATE command to the ATA device; 5) if the ATA STANDBY IMMEDIATE command in step 4 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 6) if the ATA STANDBY IMMEDIATE command in step 4 completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2). If ATA IDENTIFY DEVICE data word 49, bit 13 is set to zero or the standby timer is not enabled, then the SATL shall complete the command with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and an additional sense code of INVALID FIELD IN CDB.
All other values	The SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.	

[a] For ATA devices compliant with previous versions of ATA, ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for an ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode after completion of the ATA verify command.

9.11.2 Processing ending status if an error occurs

If an error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to zero, then the SATL shall terminate the START STOP UNIT command with CHECK CONDITION status with a sense key set to ABORTED COMMAND, and the additional sense code set to the value specified for the error being reported (see table 51).

If an error occurs during the processing of the START STOP UNIT command and the IMMED bit is set to one, then the SATL shall terminate the START STOP UNIT command and return CHECK CONDITION status as a deferred error (see SPC-4) with a sense key set to ABORTED COMMAND, and the additional sense code set to the value specified for the error being reported (see table 51).

9.11.3 START STOP UNIT START bit LOEJ bit combinations

The SATL shall perform the actions shown in table 51 in response to a START STOP UNIT command when the POWER CONDITION field is set to zero.

Table 51 — Definition of START and LOEJ bits in the START STOP UNIT CDB (part 1 of 2)

START	LOEJ	Definition
0	0	The SATL shall: 1) If the IMMED bit is set to one, then return GOOD status; 2) Send an ATA flush command (see 3.1.12) to the ATA device; 3) If the ATA flush command in step 2 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; 4) If the ATA flush command in step 2 completes without error, then send an ATA STANDBY IMMEDIATE command to the ATA device with the Count field set to zero; 5) If the ATA STANDBY IMMEDIATE in step 4 command completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to COMMAND SEQUENCE ERROR; and 6) If the ATA STANDBY IMMEDIATE in step 4 command completes without error and the IMMED bit is set to zero, then return GOOD status (see 9.11.2) [a].
0	1	If the ATA device supports the Removable Media feature set (see ATA/ATAPI-7), then the SATL shall: 1) If the IMMED bit is set to one, then return GOOD status; 2) Send an ATA MEDIA EJECT command to the ATA device; 3) If the ATA MEDIA EJECT command in step 2 completes with an error, then process ending status according to the IMMED bit (see 9.11.2) with the additional sense code set to MEDIA LOAD OR EJECT FAILED; and 4) If the MEDIA EJECT command in step 2 completes without error and the IMMED bit is set to zero, then return GOOD status. If the ATA device does not support the Removable Media feature set, then the SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

[a] After returning GOOD status for a START STOP UNIT command with the START bit set to zero, the SATL shall consider the ATA device to be in the Stopped power state (see SBC-3).

[b] For ATA devices compliant with previous versions of ATA, an ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for the ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode (see ATA8-ACS) after completion of the ATA verify command.

[c] After returning GOOD status for a START STOP UNIT command with the START bit set to one, the SATL shall consider the ATA device to be in the Active power state (see SBC-3).

Table 51 — Definition of START and LOEJ bits in the START STOP UNIT CDB (part 2 of 2)

START	LOEJ	Definition
1	0	The SATL shall: 1) If the IMMED bit is set to one, then return GOOD status; 2) Send an ATA verify command (see 3.1.24) to the ATA device with the Count field set to one and the LBA set to a value between zero and the maximum LBA supported by the ATA device in its current configuration [b]; and 3) If the IMMED bit is set to zero, then return GOOD status when command completion is received for the ATA verify command (see 3.1.24) [c].
1	1	The SATL shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST, with the additional sense code set to INVALID FIELD IN CDB.

[a] After returning GOOD status for a START STOP UNIT command with the START bit set to zero, the SATL shall consider the ATA device to be in the Stopped power state (see SBC-3).

[b] For ATA devices compliant with previous versions of ATA, an ATA device medium access occurs when an LBA is specified whose data is not contained in the ATA device's cache memory. If a value in LBA is specified for the ATA verify command where the data is contained in the ATA device's cache memory, then the ATA device may not be in the Active power mode (see ATA8-ACS) after completion of the ATA verify command.

[c] After returning GOOD status for a START STOP UNIT command with the START bit set to one, the SATL shall consider the ATA device to be in the Active power state (see SBC-3).

9.11.4 NO_FLUSH translation

The NO_FLUSH bit specifies if the SATL sends an ATA flush command while processing certain power condition requests (see table 50).

9.12 SYNCHRONIZE CACHE (10) command

The SYNCHRONIZE CACHE (10) command is used to flush the most recent data in the cache of the ATA device to physical medium.

Table 52 shows the translation for fields in the SYNCHRONIZE CACHE (10) CDB.

Table 52 — SYNCHRONIZE CACHE (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 35h. The SATL shall send an ATA flush command (see 3.1.12) in accordance with the constraints described in 9.1.
SYNC_NV	Unspecified (see 3.4.2)
IMMED	If the IMMED bit is set to one then the SATL shall return GOOD status and then send an ATA flush command. If the IMMED bit is set to zero then the SATL shall send an ATA flush command and return status upon completion.
LOGICAL BLOCK ADDRESS	The SATL shall ignore this field and shall process this command as though this field contained zero.
GROUP NUMBER	Unspecified (see 3.4.2)
NUMBER OF BLOCKS	The SATL shall ignore this field and shall process this command as though this field contained zero (see SBC-3).
CONTROL	6.5

9.13 SYNCHRONIZE CACHE (16) command

The SYNCHRONIZE CACHE (16) command is used to flush the most recent data in the cache of the ATA device to physical medium.

Table 53 shows the translation for fields in the SYNCHRONIZE CACHE (16) CDB.

Table 53 — SYNCHRONIZE CACHE (16) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 91h. The SATL shall send an ATA flush command (see 3.1.12) in accordance with the constraints described in 9.1.
SYNC_NV	As defined in SYNCHRONIZE CACHE (10) (see 9.12).
IMMED	As defined in SYNCHRONIZE CACHE (10) (see 9.12).
LOGICAL BLOCK ADDRESS	As defined in SYNCHRONIZE CACHE (10) (see 9.12).
GROUP NUMBER	As defined in SYNCHRONIZE CACHE (10) (see 9.12).
NUMBER OF BLOCKS	As defined in SYNCHRONIZE CACHE (10) (see 9.12).
CONTROL	6.5

9.14 VERIFY (10) command

The VERIFY (10) command is used to verify data on the ATA device's medium. Table 54 shows the translation of fields in the VERIFY (10) CDB.

Table 54 — VERIFY (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 2Fh.
VRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and if the BYTCHK bit is set to one, then the SATL shall perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device using an ATA read command (see 3.1.21) by the SATL, and return completion status reflecting the results of the comparison as described in SBC-3. If the BYTCHK bit is set to zero, the SATL shall send an ATA verify command (see 3.1.24) in accordance with the constraints defined in 9.1.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA verify command (see 3.1.24) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
VERIFICATION LENGTH	The verification length shall be used to set the ATA Sector Count as defined in 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA Sector Count (see 3.1.16) in the ATA verify command (see 3.1.24) equal to the value specified in the VERIFICATION LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.5

9.15 VERIFY (12) command

Table 55 shows the translation of fields in the VERIFY (12) CDB.

Table 55 — VERIFY (12) CDB field translations

Field	Description or reference
OPERATION CODE	Set to AFh. The SATL shall send an ATA verify command (see 3.1.24) in accordance with the constraints defined in 9.1.
VRPROTECT	As defined in VERIFY (10) (see 9.14).
DPO	As defined in VERIFY (10) (see 9.14).
BYTCHK	As defined in VERIFY (10) (see 9.14).
LOGICAL BLOCK ADDRESS	As defined in VERIFY (10) (see 9.14).
VERIFICATION LENGTH	As defined in VERIFY (10) (see 9.14).
GROUP NUMBER	As defined in VERIFY (10) (see 9.14).
CONTROL	6.5

9.16 VERIFY (16) command

Table 56 shows the translation of fields in the VERIFY (16) CDB.

Table 56 — VERIFY (16) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 8Fh. The SATL shall send an ATA verify command (see 3.1.24) in accordance with the constraints defined in 9.1.
VRPROTECT	As defined in VERIFY (10) (see 9.14).
DPO	As defined in VERIFY (10) (see 9.14).
BYTCHK	As defined in VERIFY (10) (see 9.14).
LOGICAL BLOCK ADDRESS	As defined in VERIFY (10) (see 9.14).
VERIFICATION LENGTH	As defined in VERIFY (10) (see 9.14).
GROUP NUMBER	As defined in VERIFY (10) (see 9.14).
CONTROL	6.5

9.17 WRITE commands overview

9.17.1 WRITE commands operation code translation

This subclause applies to the translation of the SCSI WRITE(6) command, WRITE(10) command, WRITE(12) command, and WRITE(16) command.

The SATL shall transfer the logical blocks specified in the SCSI write command (see 3.1.81) from the SCSI application client to the ATA device. The SATL shall send ATA write commands (see 3.1.26) in accordance with the constraints specified in 9.1.

9.17.2 WRITE commands with FUA

This subclause applies to the translation of the WRITE (10) command, WRITE (12) command, and WRITE (16) command.

If the FUA bit is set to zero in the SCSI write command CDB, then the SATL shall process this command as described in 9.17.1.

If the FUA bit is set to one in the SCSI write command CDB, then the SATL shall send the following, in accordance with the constraints described in 9.1:

a) the following ATA commands:
   1) an ATA write command (see 3.1.26) excluding WRITE DMA FUA EXT, WRITE DMA QUEUED FUA EXT, WRITE MULTIPLE FUA EXT, and WRITE FPDMA QUEUE; and
   2) an ATA verify command (see 3.1.24);
b) one of the following ATA commands (see ATA8-ACS):
   A) WRITE DMA FUA EXT;
   B) WRITE DMA QUEUED FUA EXT; or
   C) WRITE MULTIPLE FUA EXT;
   or
c) an ATA WRITE FPDMA QUEUED command (see SATA-2.6) with the FUA bit in the Device field set to one.

See 5.4 for a description of multiple command sequence error handling.

9.18 WRITE (6) command

The WRITE (6) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 57 shows the translation of fields in the WRITE (6) CDB.

Table 57 — WRITE (6) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 0Ah. See 9.17.1.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15) as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.26) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
TRANSFER LENGTH [a]	The transfer length shall be used to set the ATA Sector Count (see 3.1.22), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA Sector Count (see 3.1.22) in the ATA write command (see 3.1.26) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2).
CONTROL	6.5

[a] A TRANSFER LENGTH field set to zero specifies a transfer of 256 logical blocks (see SBC-3).

9.19 WRITE (10) command

The WRITE (10) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 58 shows the translation of fields in the WRITE (10) CDB.

Table 58 — WRITE (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 2Ah. See 9.17.1 and 9.17.2.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
FUA	9.17.2
FUA_NV	The SATL may ignore the FUA_NV bit or the SATL may implement the FUA_NV bit as defined in SBC-3.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.26) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH[a]	The transfer length is used to set the ATA Sector Count (see 3.1.22), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.26) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall send as many ATA write commands (see 3.1.26) as needed to satisfy the transfer length specified by the WRITE (10) command.
CONTROL	6.5

[a] A transfer length of zero specifies that a data transfer shall not take place.

9.20 WRITE (12) command

The WRITE (12) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 59 shows the translation of fields in the WRITE (12) CDB.

Table 59 — WRITE (12) CDB field translations

Field	Description or reference
OPERATION CODE	Set to AAh. See 9.17.1 and 9.17.2.
WRPROTECT	As defined in WRITE (10) (see 9.19).
DPO	As defined in WRITE (10) (see 9.19).
FUA	As defined in WRITE (10) (see 9.19).
FUA_NV	As defined in WRITE (10) (see 9.19).
LOGICAL BLOCK ADDRESS	As defined in WRITE (10) (see 9.19).
TRANSFER LENGTH[a]	As defined in WRITE (10) (see 9.19).
GROUP NUMBER	As defined in WRITE (10) (see 9.19).
CONTROL	6.5
[a] A transfer length of zero specifies that a data transfer shall not take place.	

9.21 WRITE (16) command

The WRITE (16) command is used to request the SATL to transfer user data from the application client to the ATA device. Data may be written to the medium or to the cache of the ATA device.

Table 60 shows the translation of fields in the WRITE (16) CDB.

Table 60 — WRITE (16) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 8Ah. See 9.17.1 and 9.17.2.
WRPROTECT	As defined in WRITE (10) (see 9.19).
DPO	As defined in WRITE (10) (see 9.19).
FUA	As defined in WRITE (10) (see 9.19).
FUA_NV	As defined in WRITE (10) (see 9.19).
LOGICAL BLOCK ADDRESS	As defined in WRITE (10) (see 9.19).
TRANSFER LENGTH	As defined in WRITE (10) (see 9.19).
GROUP NUMBER	As defined in WRITE (10) (see 9.19).
CONTROL	6.5

9.22 WRITE AND VERIFY commands overview

This subclause applies to the translation of the WRITE AND VERIFY (10) command, WRITE AND VERIFY (12) command, and WRITE AND VERIFY (16) command.

The SATL shall send:

1) an ATA write command (see 3.1.26) in accordance with the constraints defined in 9.1; and
2) an ATA verify command (see 3.1.24).

9.23 WRITE AND VERIFY (10) command

The WRITE AND VERIFY (10) command requests that the SATL to transfer the specified logical blocks from the application client to the ATA device, and then verify that the data was written correctly to the medium of the ATA device.

Table 61 shows the translation of fields in the WRITE AND VERIFY (10) CDB.

Table 61 — WRITE AND VERIFY (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 2Eh. See 9.22.
WRPROTECT	Unspecified (see 3.4.2)
DPO	Unspecified (see 3.4.2)
BYTCHK	If the SATL supports a BYTCHK bit set to one and the BYTCHK bit is set to one, then after writing the data to the medium the SATL shall read the data from the medium and perform a byte-by-byte comparison of the data transferred from the application client to the SATL with data read from the ATA device using an ATA read command (see 3.1.21), and then shall return completion status reflecting the results of the comparison as described in SBC-3. If the BYTCHK bit is set to zero, the SATL shall send an ATA verify command (see 3.1.24) in accordance with the constraints defined in 9.1.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA write command (see 3.1.26) and the ATA verify command (see 3.1.24) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
TRANSFER LENGTH[a]	The transfer length is used to set the ATA Sector Count (see 3.1.22), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the ATA Sector Count in the ATA write command (see 3.1.26) and the ATA verify command (see 3.1.24) equal to the value specified in the TRANSFER LENGTH field. Otherwise, the mapping is unspecified (see 3.4.2). The SATL shall send as many ATA write commands and ATA verify commands as needed to satisfy the transfer length specified by the WRITE AND VERIFY (10) command.
CONTROL	6.5

[a] A transfer length of zero specifies that a data transfer shall not take place.

9.24 WRITE AND VERIFY (12) command

The WRITE AND VERIFY (12) command requests that the SATL to transfer the specified logical blocks from the application client to the ATA device, and then verify that the data was written correctly to the medium of the ATA device.

Table 62 shows the translation of fields in the WRITE AND VERIFY (12) CDB.

Table 62 — WRITE AND VERIFY (12) CDB field translations

Field	Description or reference
OPERATION CODE	Set to AEh. See 9.22.
WRPROTECT	As defined in WRITE AND VERIFY (10) (see 9.23).
DPO	As defined in WRITE AND VERIFY (10) (see 9.23).
BYTCHK	As defined in WRITE AND VERIFY (10) (see 9.23).
LOGICAL BLOCK ADDRESS	As defined in WRITE AND VERIFY (10) (see 9.23).
TRANSFER LENGTH[a]	As defined in WRITE AND VERIFY (10) (see 9.23).
GROUP NUMBER	As defined in WRITE AND VERIFY (10) (see 9.23).
CONTROL	6.5
[a] A transfer length of zero specifies that a data transfer shall not take place.	

9.25 WRITE AND VERIFY (16) command

The WRITE AND VERIFY (16) command requests that the SATL to transfer the specified logical blocks from the application client to the ATA device, and then verify that the data was written correctly to the medium of the ATA device.

Table 63 shows the translation of fields in the WRITE AND VERIFY (16) CDB.

Table 63 — WRITE AND VERIFY (16) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 8Eh. See 9.22.
WRPROTECT	As defined in WRITE AND VERIFY (10) (see 9.23).
DPO	As defined in WRITE AND VERIFY (10) (see 9.23).
BYTCHK	As defined in WRITE AND VERIFY (10) (see 9.23).
LOGICAL BLOCK ADDRESS	As defined in WRITE AND VERIFY (10) (see 9.23).
GROUP NUMBER	As defined in WRITE AND VERIFY (10) (see 9.23).
TRANSFER LENGTH[a]	As defined in WRITE AND VERIFY (10) (see 9.23).
CONTROL	6.5
[a] A transfer length of zero specifies that a data transfer shall not take place.	

9.26 WRITE LONG (10) command

The WRITE LONG (10) command (see SBC-3) requests that the SATL mark a logical block or physical block as containing an error.

Table 64 shows the translation of fields in the WRITE LONG (10) CDB.

Table 64 — WRITE LONG (10) CDB field translations

Field	Description or Reference
OPERATION CODE	Set to 3Fh. If the ATA device does not support the ATA WRITE UNCORRECTABLE EXT command (see ATA8-ACS), then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID OPERATION CODE.
COR_DIS	See table 65.
WR_UNCOR	See table 65.
PBLOCK	See table 65.
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15) as defined by 9.1. If the SATL implements direct block mapping (see 3.1.35), then the SATL shall set the ATA LBA in the ATA WRITE UNCORRECTABLE EXT command equal to the value specified in the logical block address field. Otherwise the mapping is unspecified (see 3.4.2).
BYTE TRANSFER LENGTH	If the BYTE TRANSFER LENGTH field is not set to zero, the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
CONTROL	6.5

The interaction of the WR_UNCOR bit and the PBLOCK bit are defined in table 65.

Table 65 — WR_UNCOR bit and PBLOCK bit

COR_DIS	WR_UNCOR	PBLOCK	Description
0	1	0	If the ATA logical sectors per physical sector exponent is non-zero, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. If the ATA logical sectors per physical sector exponent is zero, then the SATL shall send a ATA WRITE UNCORRECTABLE EXT command with: a) the Feature field set to 55h (i.e, pseudo-uncorrectable error with logging); and b) the Count field set to 0001h.
0	1	1	The SATL shall send an ATA WRITE UNCORRECTABLE EXT command with: a) the Feature field set to 55h (i.e., pseudo-uncorrectable error with logging); and b) the Count field set to 0001h.
1	1	0	The SATL shall send an ATA WRITE UNCORRECTABLE EXT command with: a) the Feature field set to AAh (i.e., flagged error without logging); and b) the Count field set to 0001h.
All others			The SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

9.27 WRITE LONG (16) command

The WRITE LONG (16) command (see SBC-3) requests that the SATL mark a logical block or physical block as containing an error.

Table 66 shows the translation of fields in the WRITE LONG (16) CDB.

Table 66 — WRITE LONG (16) CDB field translations

Field	Description or Reference
OPERATION CODE / SERVICE ACTION	Set to 9Fh / 11h.
COR_DIS	As defined in WRITE LONG (10) (see 9.26)
WR_UNCOR	As defined in WRITE LONG (10) (see 9.26)
PBLOCK	As defined in WRITE LONG (10) (see 9.26)
LOGICAL BLOCK ADDRESS	As defined in WRITE LONG (10) (see 9.26)
BYTE TRANSFER LENGTH	As defined in WRITE LONG (10) (see 9.26)

9.28 WRITE SAME (10) command

9.28.1 WRITE SAME (10) command overview

The WRITE SAME (10) command requests that the SATL transfer a single logical block from the application client and write the contents of that single logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses on the ATA device.

Table 67 shows the translations of the fields in the WRITE SAME (10) CDB.

Table 67 — WRITE SAME (10) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 41h. If the ATA device supports the ATA SCT Write Same command (see ATA8-ACS), the SATL should send the ATA SCT Write Same command to repeatedly write the data block transferred from the application client to the ATA device. If the ATA device does not implement ATA SCT Write Same command then the SATL shall send ATA write commands as defined in 9.17.1.
WRPROTECT	Unspecified (see 3.4.2)
PBDATA	9.28.2
LBDATA	9.28.2
LOGICAL BLOCK ADDRESS	The logical block address shall be used to set the ATA LBA (see 3.1.15), as defined by 9.1. If the SATL implements direct logical block mapping (see 3.1.35), then the SATL shall set the Start in the ATA SCT Write Same command or the ATA LBA in the ATA write command (see 3.1.26) equal to the value specified in the LOGICAL BLOCK ADDRESS field. Otherwise, the mapping is unspecified (see 3.4.2).
GROUP NUMBER	Unspecified (see 3.4.2)
NUMBER OF LOGICAL BLOCKS	A NUMBER OF LOGICAL BLOCKS field set to zero specifies that the SATL shall repeatedly write the logical block transferred from the application client to the range of ATA logical sectors corresponding to the range of logical block addresses specified by the LOGICAL BLOCK ADDRESS field through the LBA of the last logical block on the logical unit. If the NUMBER OF LOGICAL BLOCKS field is set to a value other than zero, then the SATL shall repeatedly write the data block transferred from the application client to the medium of the ATA device for the number of logical blocks specified to the corresponding logical sectors on the ATA device. The SATL shall send as many ATA commands as required to satisfy the number of blocks specified by the WRITE SAME (10) command.
CONTROL	6.5

9.28.2 LBDATA bit and PBDATA bit

The SATL shall write data to the specified logical block addresses according to the values in the LBDATA and PBDATA bits as shown in table 68.

Table 68 — LBDATA and PBDATA fields

LBDATA	PBDATA	Description
0	0	The SATL shall write the block of data transferred from the application client to the range of blocks specified in the LOGICAL BLOCK ADDRESS field and the NUMBER OF LOGICAL BLOCKS field, repeatedly, on the medium of the ATA device. If the ATA device supports the ATA SCT Write Same command, then the SATL should use the ATA SCT Write Same command with the Function Code set to 0002h or 0004h for writing the data. Otherwise, the SATL shall use ATA write commands as defined in 9.17.2 (see SBC-3).
1	0	The SATL shall replace the first four bytes of the logical block received from the application client with the least significant four bytes of the LBA of the logical block being written to the media, ending with the least significant byte (e.g., if the LBA is 7766_5544_3322_1100h, 3322_1100h is written with 33h written first and 00h written last). The SATL shall use ATA write commands as defined in 9.17.2 (see SBC-3).
0	1	The SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
1	1	

9.29 WRITE SAME (16) command

The WRITE SAME (16) command (see table 69) requests that the SATL transfer a single logical block from the application client and write the contents of that single logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses on the ATA device.

Table 69 shows the translations of the fields in the WRITE SAME (10) CDB.

Table 69 — WRITE SAME (16) CDB field translations

Field	Description or reference
OPERATION CODE	Set to 93h. If the ATA device supports the ATA SCT Write Same command (see ATA8-ACS), then the SATL should send the ATA SCT Write Same command to repeatedly write the data block transferred from the application client to the ATA device. If the ATA device does not implement ATA SCT Write Same command, then the SATL shall send ATA write commands as defined in 9.17.1.
WRPROTECT	As defined in WRITE SAME (10) (see 9.28).
LBDATA	As defined in WRITE SAME (10) (see 9.28).
PBDATA	As defined in WRITE SAME (10) (see 9.28).
LOGICAL BLOCK ADDRESS	As defined in WRITE SAME (10) (see 9.28).
NUMBER OF LOGICAL BLOCKS	As defined in WRITE SAME (10) (see 9.28).
GROUP NUMBER	As defined in WRITE SAME (10) (see 9.28).
CONTROL	6.5

10 Parameters for SAT implementations

10.1 Mode parameters

10.1.1 General information

SCSI mode parameters provide a mechanism to set operating parameters for SCSI devices and logical units. The MODE SENSE command obtains operating parameters and the MODE SELECT command sets operating parameters. This standard does not define the content of most operating parameters defined in mode pages due to lack of equivalent operations or features defined for ATA devices. The SATL emulates a SCSI device server for all MODE SENSE and MODE SELECT commands, and emulates the mode pages listed in 10.1.2.

The Mode Page Policy VPD page (see 10.3) should be implemented. If implemented, then the MODE PAGE POLICY field in each mode page policy descriptor should be set to 00b (i.e., shared) for each mode page and only one copy of mode page values should be maintained for all logical units within a target device (i.e., the MLUS bit is set to one in each mode page policy descriptor).

If the Mode Page Policy VPD page is not implemented, then the SATL shall maintain shared mode pages for all I_T nexuses and shall share mode pages across all logical units within a target device.

10.1.2 Commonly used SCSI mode pages overview

This standard defines translations for the mode pages listed in table 70.

Table 70 — Summary of SCSI / ATA mode page mapping

SCSI mode page	Reference
Mode parameter header	10.1.3
Mode parameter block descriptor	10.1.4
Control (i.e., 0Ah)	10.1.5
Read-Write Error Recovery (i.e., 01h)	10.1.6
Caching (i.e., 08h)	10.1.7
Informational Exceptions Control (i.e., 1Ch)	10.1.8
All others	See SPC-4 and SBC-3 Unspecified (see 3.4.2)

10.1.3 Mode parameter headers

Table 71 shows the fields in the mode parameter header for the MODE SELECT (6) command and the MODE SENSE (6) command.

Table 71 — Mode parameter header (6) fields

Field	Description or reference
MODE DATA LENGTH	Unspecified (see 3.4.2)
MEDIUM TYPE	This field should be set to 00h. When processing a MODE SELECT command, if the MEDIUM TYPE field is set to a value other than 00h, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
DEVICE SPECIFIC PARAMETER	Unspecified (see 3.4.2) for the MODE SELECT command. For the MODE SENSE command, the DEVICE SPECIFIC PARAMETER field for direct-access block devices contains the DPOFUA bit and the WP bit (see SBC-3). A DPOFUA bit set to zero indicates that the SATL supports neither the DPO bit nor the FUA bit. A DPOFUA bit set to one indicates that the SATL supports both the DPO bit and the FUA bit. A WP bit set to zero indicates that the medium is not write-protected. A WP bit set to one indicates that the medium is write-protected.
BLOCK DESCRIPTOR LENGTH	This value is obtained by multiplying the number of block descriptors by eight (see SPC-4). The SATL shall support zero or one mode parameter block descriptors.

Table 72 shows the fields in the mode parameter header for the MODE SELECT (10) command and the MODE SENSE (10) command.

Table 72 — Mode parameter header (10) fields

Field	Description or reference
MODE DATA LENGTH	(see table 71)
MEDIUM TYPE	(see table 71)
DEVICE SPECIFIC PARAMETER	(see table 71)
LONGLBA	Describes the length of the block descriptors as follows: a) If set to zero, then the mode parameter block descriptor is eight bytes long; or b) If set to one, then the mode parameter block descriptor is 16 bytes long.
BLOCK DESCRIPTOR LENGTH	This field specifies (i.e., for a MODE SELECT command) or indicates (i.e., for a MODE SENSE command) the length of the mode parameter block descriptor. This value is obtained by multiplying the number of block descriptors by eight if LONGLBA bit is set to zero or by 16 if LONGLBA bit is set to one. The SATL shall support zero or one mode parameter block descriptors.

10.1.4 Mode parameter block descriptor fields

The SATL may support the direct-access short LBA mode parameter block descriptor or the long LBA mode parameter block descriptor. Table 73 describes the translation of fields in the short LBA mode parameter block descriptor and the long LBA mode parameter block descriptor supported by the SATL.

Table 73 — Mode parameter block descriptor fields

Field	Description or reference
NUMBER OF BLOCKS[a]	Unspecified (see 3.4.2)
BLOCK LENGTH[a]	When processing a MODE SELECT command, if the SATL implements direct logical block mapping (see 3.1.35) and the value of the BLOCK LENGTH field is not the same as the ATA logical sector size (see 3.1.16), then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the SATL supports indirect logical block mapping then this field is unspecified (see 3.4.2). When processing the MODE SENSE command, if the SATL implements direct logical block mapping (see 3.1.35) then the SATL shall return the same block length for the entire logical unit and the BLOCK LENGTH field shall contain the ATA logical sector size (see 3.1.16). Otherwise the BLOCK LENGTH field is unspecified (see 3.4.2).

[a] The values reported in the NUMBER OF BLOCKS field and the BLOCK LENGTH field shall be such that the logical unit capacity (see 3.1.51) is less than or equal to the ATA device capacity (see 3.1.10).

10.1.5 Control mode page

10.1.5.1 General translation

The Control mode page provides controls and information about behavior of the emulated SCSI device.

Table 74 describes the translation of the Control mode page for an ATA device.

Table 74 — Control mode page fields

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
SPF	no	Shall be set to zero.
PAGE CODE	no	Shall be set to 0Ah.
PAGE LENGTH	no	Shall be set to 0Ah.
TST	no	Shall be set to 000b to indicate that a SCSI representation of an ATA device has one task set for all initiators
TMF_ONLY	no	Shall be set to zero[a]
D_SENSE	Unspecified	The SATL shall support this bit as defined in SPC-4 with the following exception: a) if the D_SENSE bit is set to zero (i.e., fixed format sense data), the SATL should return fixed format sense data for ATA PASS-THROUGH commands.[b]
GLTSD	Unspecified	Unspecified (see 3.4.2)
RLEC	no	Shall be set to zero
QUEUE ALGORITHM MODIFIER	no	The QUEUE ALGORITHM MODIFIER bit shall be set to one.
QERR	no	If the SATL supports the full task management model and ATA abort retry (see 3.1.7) of ATA queued commands (see 3.1.20) aborted by ATA collateral abort (see 3.1.8), then the SATL shall set this field to 00b. Otherwise, the SATL shall set this field to 01b and comply with the unit attention condition requirements for a command completed with CHECK CONDITION status (see SPC-4).
RAC	Unspecified	Unspecified (see 3.4.2)
UA_INTLCK_CTRL	no	Shall be set to 00b
SWP	no	Shall be set to zero
ATO	Unspecified	Unspecified (see 3.4.2)
TAS	no	Shall be set to zero
AUTOLOAD MODE	no	Shall be set to 000b
BUSY TIMEOUT PERIOD	Unspecified	The default value shall be set to FFFFh. A SATL may support variable timeout periods and allow the application client to set a new value through a MODE SELECT operation for this mode page (see SPC-4).
EXTENDED SELF-TEST COMPLETION TIME	no	10.1.5.2

[a] SATL implementations shall not support ACA.
[b] SATLs compliant with previous versions of this standard return descriptor format sense data for the ATA PASS-THROUGH commands regardless of the value of the D_SENSE bit.

10.1.5.2 Extended self-test completion time

A SATL implementation shall set the EXTENDED SELF-TEST COMPLETION TIME field to 0000h unless the ATA device supports SMART self-tests and the SATL supports a value other than 000b for the SELF-TEST CODE field for a SEND DIAGNOSTIC command. The SATL determines if the ATA device supports SMART self-test by

examining the value of ATA IDENTIFY DEVICE data word 84 bit 1. If ATA IDENTIFY DEVICE data word 84 bit 1 is set to one, then the ATA device supports the SMART self-test and shall retrieve the ATA device SMART data structure from the ATA device by sending an ATA SMART READ DATA command to the ATA device. The SATL may cache the ATA SMART READ data for future use when a subsequent MODE SENSE command requests the control mode page. If the SATL caches such data, the SATL may reference the cached copy instead of sending a new ATA SMART READ DATA command, then the SATL shall set the EXTENDED SELF-TEST COMPLETION TIME field as follows:

1) If byte 373 of the returned SMART data structure is not set to FFh, then the SATL shall set the EXTENDED SELF-TEST COMPLETION TIME field to a value that is 60 times the contents of byte 373; or
2) If byte 373 of the returned SMART data structure is set to FFh, then the SATL shall set the EXTENDED SELF-TEST COMPLETION TIME field to a value that is the lesser of FFFFh or the result of the following formula:

EXTENDED SELF-TEST COMPLETION TIME field = ((w x 256) + z) x 60

where:

w is the contents of byte 376; and

z is the contents of byte 375.

10.1.6 Read-Write Error Recovery mode page

The Read-Write Error Recovery mode page specifies the error recovery parameters the SATL shall use during a command that performs a read or write operation to the medium of the ATA device (see SBC-3). Table 75 defines the translation for the Read-Write Error Recovery mode page.

Table 75 — Read-Write Error Recovery mode page fields

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
SPF	no	Shall be set to zero
PAGE CODE	no	Shall be set to 01h
PAGE LENGTH	no	Shall be set to 0Ah
AWRE	no	Shall be set to one (see SBC-3)
ARRE	no	Shall be set to zero (see SBC-3)
TB	n/a	Unspecified (see 3.4.2)
RC	no	Shall be set to zero (see SBC-3)
EER	no	Shall be set to zero (see SBC-3)
PER	no	Shall be set to zero (see SBC-3)
DTE	no	Shall be set to zero (see SBC-3)
DCR	no	Shall be set to zero (see SBC-3)
READ RETRY COUNT	n/a	Unspecified (see 3.4.2)
WRITE RETRY COUNT	n/a	Unspecified (see 3.4.2)
RECOVERY TIME LIMIT	no	Shall be set to zero (see SBC-3)

10.1.7 Caching mode page

The Caching mode page defines parameters that affect the behavior of the cache in the ATA device.

Table 76 shows the translation of fields in the Caching mode page.

Table 76 — Caching mode page fields (part 1 of 2)

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
SPF	no	Shall be set to zero
PAGE CODE	no	Shall be set to 08h
PAGE LENGTH	no	Shall be set to 12h
IC	no	Shall be set to zero
ABPF	no	Shall be set to zero
CAP	no	Shall be set to zero
DISC	no	Shall be set to zero
SIZE	no	Shall be set to zero
WCE	yes[a]	When processing a MODE SENSE command, the SATL shall determine if the write cache of the ATA device is enabled from the ATA IDENTIFY DEVICE data word 85 bit 5. If the write cache of the ATA device is enabled then the SATL shall return a value of one for the WCE bit. If the write cache of the ATA device is disabled then the SATL shall return a value of zero for the WCE bit. When processing a MODE SELECT command: a) if the WCE bit is set to zero, then the SATL shall disable the write cache of the ATA device by issuing an ATA SET FEATURES – Disable write cache command (i.e., with the Features register set to 82h); or b) if the WCE bit is set to one, then the SATL shall enable the write cache of the ATA device by issuing an ATA SET FEATURES – Enable write cache command (i.e., with the Features register set to 02h).
MF	no	Shall be set to zero
RCD	no	Shall be set to zero
DEMAND READ RETENTION PRIORITY	no	Shall be set to zero
WRITE RETENTION PRIORITY	no	Shall be set to zero
DISABLE PRE-FETCH TRANSFER LENGTH	no	Shall be set to zero
MINIMUM PRE-FETCH	no	Shall be set to zero
MAXIMUM PRE-FETCH	no	Shall be set to zero
MAXIMUM PRE-FETCH CEILING	no	Shall be set to zero
FSW	no	Shall be set to zero

[a] If the ATA device does not support a write cache (i.e., ATA IDENTIFY DEVICE data, word 85 bit 5 is set to zero), this field is not changeable.

Table 76 — Caching mode page fields (part 2 of 2)

Field	Changeable	Description or reference
LBCSS	no	Shall be set to zero
DRA	yes	When processing a MODE SENSE command, the SATL shall determine if the ATA device look-ahead is enabled from the ATA IDENTIFY DEVICE data word 85 bit 6. If the look-ahead is enabled then the SATL shall return a value of zero for the DRA bit. If the look-ahead is disabled then the SATL shall return a value of one for the DRA bit. When processing a MODE SELECT command: a) if the DRA bit is set to zero, the SATL shall enable the ATA device read look-ahead feature by issuing an ATA SET FEATURES – Enable read look-ahead feature command (i.e., with the Features register set to AAh); or b) if the DRA bit is set to one, the SATL shall disable the ATA device read look-ahead feature by issuing an ATA SET FEATURES – Disable read look-ahead feature command (i.e., with the Features register set to 55h).
NV_DIS	no	Shall be set to zero
NUMBER OF CACHE SEGMENTS	no	Shall be set to zero
CACHE SEGMENT SIZE	no	Shall be set to zero

[a] If the ATA device does not support a write cache (i.e., ATA IDENTIFY DEVICE data, word 85 bit 5 is set to zero), this field is not changeable.

10.1.8 Informational Exceptions Control mode page

10.1.8.1 Informational Exceptions Control mode page overview

The Informational Exceptions Control mode page defines the methods used by the SATL to control the reporting and the operations of specific informational exception conditions. The Informational Exceptions Control mode page applies to informational exceptions that return an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the application client (see SPC-4).

Table 77 shows the translation of fields in the Informational Exceptions Control mode page.

Table 77 — Informational Exceptions Control mode page fields

Field	Changeable	Description or reference
PS	n/a	Unspecified (see 3.4.2)
SPF	no	Shall be set to zero
PAGE CODE	no	Shall be set to 1Ch. The SATL shall determine if the ATA SMART feature set is supported from the ATA IDENTIFY DEVICE data word 82 bit 0. If the ATA SMART feature set is not supported, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB for a MODE SENSE command or INVALID FIELD IN PARAMETER LIST for a MODE SELECT command.
PAGE LENGTH	no	Shall be set to 0Ah
PERF	no	Shall be set to zero
EBF	n/a	Unspecified (see 3.4.2)
EWASC	n/a	Unspecified (see 3.4.2)
DEXCPT	yes	Unspecified (see 3.4.2)
TEST	no	Shall be set to zero
EBACKERR	no	Unspecified (see 3.4.2)
LOGERR	n/a	Unspecified (see 3.4.2)
MRIE	no[a]	Should be set to 6h (see 10.1.8.2).
INTERVAL TIMER	n/a	Unspecified (see 3.4.2)
REPORT COUNT	n/a	Unspecified (see 3.4.2)

[a] The MRIE field should be set to 6h, however if the SATL supports other settings of the MRIE field, the SATL should permit the MRIE field to be changeable.

10.1.8.2 Method of reporting informational exceptions (MRIE)

The SATL should support 6h. Support for any other value is unspecified (see 3.4.2).

When the MRIE field is set to 6h and the SATL receives a REQUEST SENSE command, the SATL shall send an ATA SMART RETURN STATUS command to the ATA device and return status to the application client as defined in SPC-4 (see 10.2.6.2). If the result of the ATA SMART RETURN STATUS command indicates a threshold exceeded condition, then the SATL shall set the additional sense code to HARDWARE IMPENDING FAILURE GENERAL HARD DRIVE FAILURE.

10.1.9 Power Condition mode pages

10.1.9.1 Power Condition mode pages overview

The Power Condition mode pages allow changing of the ATA APM mode setting and the ATA STANDBY timer value. They also provide information about the current power condition settings.

10.1.9.2 ATA Power Condition mode page

The ATA Power Condition Mode page is ATA specific and defined in 12.3.3.

10.1.9.3 Power Condition mode page

The Power Condition mode page translation (see table 78) allows setting and examining the ATA STANDBY timer value (see ATA8-ACS). Values in the STANDBY TIMER field for the MODE SENSE command shall be translated as defined in table 79.

Table 78 — Power Condition mode page fields

Field	Changeable	Description or Reference
PS	n/a	Unspecified (see 3.4.2)
SPF	no	Shall be set to zero
PAGE CODE	no	Shall be set to 1Ah
PAGE LENGTH	no	Shall be set to 0Ah
IDLE	no	When processing a MODE SENSE command, the IDLE bit shall be returned as zero. When processing a MODE SELECT command, if the IDLE bit is set to one, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
STANDBY	yes	When processing a MODE SENSE command, if ATA IDENTIFY DEVICE data word 49, bit 13 is set to one and the ATA standby timer is enabled, then the STANDBY bit shall be returned as one. If ATA IDENTIFY DEVICE data word 49, bit 13 is set to zero or the ATA standby timer is not enabled, then the STANDBY bit shall be returned as zero. When processing a MODE SELECT command, if the STANDBY bit is set to one, then: 1) If the ATA IDENTIFY DEVICE data word 49, bit 13 is set to zero, then the SATL shall terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST; and 2) If the ATA IDENTIFY DEVICE data word 49, bit 13 is set to one, then the SATL shall send the ATA STANDBY command to the ATA device and the value in the STANDBY CONDITION TIMER field shall be translated as defined in table 80 and used to set the Timer period value (TPV) (i.e., ATA Count field).
IDLE CONDITION TIMER	no	When processing a MODE SENSE command, this field shall be returned as zero. When processing a MODE SELECT command, this field shall be ignored.
STANDBY CONDITION TIMER	yes	When processing a MODE SENSE command, if the ATA IDENTIFY DEVICE data word 49, bit 13 is set to zero, then this field shall be set to zero. If the ATA IDENTIFY DEVICE data word 49, bit 13 is set to one, then the ATA standby timer value shall be translated as defined in table 79 and returned in this field. When processing a MODE SELECT command, if the STANDBY bit is set to one, then the value in this field shall be translated as defined in table 80 and used to set the Timer period value (TPV) (i.e., ATA Count field). The SATL may retain this value for return when processing a MODE SENSE command.

[a] If ATA IDENTIFY DEVICE data word 49, bit 13 is set to zero indicating that the ATA standby timer is managed by the device, the standby timer is not considered enabled and cannot be set by this MODE SELECT.

Table 79 — MODE SENSE STANDBY TIMER field translation

ATA Count field	Time	STANDBY CONDITION TIMER field
01h through F0h	5 seconds to 1 200 seconds	ATA Count field x 50
FCh	21 minutes	12 600
FFh	21 minutes, 15 seconds	12 750
F1h through FBh	30 minutes to 330 minutes	(ATA Count field - 240) x 18 000
FDh	8 hours to 12 hours	432 000
Not retained by the SATL	n/a	FFFF_FFFFh
All other values are unspecified (see 3.4.2)		

Values in the STANDBY TIMER field for the MODE SELECT command shall be translated as defined in table 80

Table 80 — MODE SELECT STANDBY TIMER field translation

STANDBY CONDITION TIMER field	ATA Count field
0	The SATL shall send an ATA STANDBY IMMEDIATE command to the ATA device
1 to 12 000	INT((z - 1) / 50) + 1[a]
12 001 to 12 600	FCh
12 601 to 12 750	FFh
12 751 to 17 999	F1h
18 000 to 198 000	INT(z / 18 000) + 240[a]
All other values	FDh
Key: z = Contents of the Power Condition mode page STANDBY CONDITION TIMER field.	
[a] INT() is the integer result of the specified division operation with any decimal remainder discarded.	

10.2 Log parameters

10.2.1 Log parameters overview

This standard defines translations for the log pages listed in table 81.

Table 81 — Summary of SCSI / ATA log page mapping

SCSI log page	Reference
Application Client (i.e., page code 0Fh/00h)	10.2.2
Supported Log Pages (i.e., page code 00h/00h)	10.2.3
Supported Log Pages and Subpages (i.e., page code 00h/FFh)	10.2.4
Self-Test Results (i.e., page code 10h/00h)	10.2.5
Informational Exceptions (i.e., page code 2Fh/00h)	10.2.6
All others	Unspecified (see 3.4.2)

10.2.2 Application Client log page

10.2.2.1 Translation Overview

The Application Client log page provides a location for application clients to store information. A SATL translates a LOG SELECT command or LOG SENSE command to the application client log page to accesses to the ATA host vendor-specific log pages. Table 82 describes the translation of the general usage application client parameter data for the application client log page.

The SATL determines if the attached ATA device supports host vendor specific log pages by reading ATA log page address 00h using an ATA READ LOG EXT command, ATA READ LOG DMA EXT command, or ATA SMART READ LOG command.

If the attached ATA device:

a) Does not support the general purpose logging feature set and the SMART feature set is disabled; or
b) does not support host vendor-specific log pages,

then the SATL shall complete the LOG SENSE command or LOG SELECT command for the application client specific log page with a CHECK CONDITION status, a sense key of ILLEGAL REQUEST, and an additional sense code of INVALID FIELD IN CDB.

Table 82 — General usage application client parameter data fields

Field	Description or Reference
PARAMETER CODE	10.2.2.2
DU	Shall be set to one
TSD	Shall be set to zero
ETC	Shall be set to zero
TMC	This field is ignored
FORMAT AND LINKING	Shall be set to 11b
PARAMETER LENGTH	Shall be set to FCh
GENERAL USAGE PARAMETER BYTES	10.2.2.2

10.2.2.2 LOG SELECT translation

The SATL stores the application client parameter for a LOG SELECT command in the ATA device host vendor-specific log page. The SATL stores the application client parameter data at the ATA log address as specified in table 83.

Within an ATA log address, the SATL shall store each parameter code in ascending order within the sixteen 512-byte data blocks for each ATA log address (e.g., parameter code 0000h is stored at offset 0 of the first 512-byte block of data at log address 90h and parameter code 0001h is stored at offset 256 in the first 512-byte block of data at log address 90h). The SATL stores this information by issuing an ATA SMART WRITE LOG command, ATA WRITE LOG EXT command, or ATA WRITE LOG DMA EXT command to the ATA device.

The SATL shall ensure that any previously stored data at the log address is preserved when writing to the log address for the requested parameter data.

Table 83 — Parameter Storage Location

Parameter Code	ATA Log Address
0000h through 001Fh	90h
0020h through 003Fh	91h
0040h through 005Fh	92h
0060h through 007Fh	93h
0080h through 009Fh	94h
00A0h through 00BFh	95h
00C0h through 00DFh	96h
00E0h through 00FFh	97h
0100h through 011Fh	98h
0120h through 013Fh	99h
0140h through 015Fh	9Ah
0160h through 017Fh	9Bh
0180h through 019Fh	9Ch
01A0h through 01BFh	9Dh
01C0h through 01DFh	9Eh
01E0h through 01FFh	9Fh
All other values	Reserved

10.2.2.3 LOG SENSE translation

The SATL retrieves the requested parameter data by reading the ATA log address that stores the parameter code using an ATA SMART READ LOG command, ATA READ LOG EXT command, or ATA READ LOG DMA EXT command. The log address to read is determined by table 83.

10.2.3 Supported Log Pages log page

The Supported Log Pages log page (see table 84) returns the list of log pages supported by the SATL (see SPC-4).

Table 84 — Supported Log Pages log page fields

Field	Description or reference
PAGE CODE	Shall be set to zero
SPF	Shall be set to zero
DS	Unspecified (see 3.4.2)
SUBPAGE CODE	Shall be set to zero
PAGE LENGTH	Unspecified (see 3.4.2)
Supported pages	The SATL shall include log pages as follows: a) the Informational Exceptions log page if the ATA device supports the ATA SMART feature set (i.e., ATA IDENTIFY DEVICE data word 82 bit 0 is set to one); and b) the Self-Test Results log page if the ATA device supports the ATA SMART self-test (i.e., ATA IDENTIFY DEVICE data word 84 bit 1 is set to one). The SATL may include other pages.

10.2.4 Support Log Pages and Subpages log page

The Supported Log Pages and Subpages log page returns the list of log pages and subpages supported by the SATL (see SPC-4).

Table 85 — Supported Log Pages and Subpages log page fields

Field	Description or reference
PAGE CODE	Shall be set to zero
SPF	Shall be set to one
DS	Unspecified (see 3.4.2)
SUBPAGE CODE	Shall be set to FFh
PAGE LENGTH	Unspecified (see 3.4.2)
Supported pages and subpages	As defined for Supported Log Pages translation (see 10.2.3)

10.2.5 Self-Test Results log page

10.2.5.1 Self-Test Results log page overview

The Self-Test Results log page provides the results from self-test results descriptor entry pointed to by the Self-test descriptor index. Table 86 shows the Self-Test Results log page header fields.

Table 86 — Self-Test Results log page fields

Field	Description or reference
PAGE CODE	Shall be set to 10h
SPF	Shall be set to zero
DS	Unspecified (see 3.4.2)
SUBPAGE CODE	Shall be set to zero
PAGE LENGTH	Shall be set to 190h

Translations of the fields for the Self-Test Results log parameters for the Self-Test Results log page are shown in table 87..

Table 87 — Self-Test Results log parameters (part 1 of 4)

Field	Description or reference
PARAMETER CODE	The SATL shall return log parameters with the PARAMETER CODE field set to 0001h through 0014h.
DU	Shall be set to zero
TSD	Shall be set to zero
ETC	Shall be set to zero
TMC	Shall be set to zero
FORMAT AND LINKING	Shall be set to 11b
PARAMETER LENGTH	Shall be set to 10h
SELF-TEST CODE	The SATL shall read the ATA log data as defined in 10.2.5.2. If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first log page (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the SELF-TEST CODE field to zero for each of the log parameters returned. If the value contained in the Self-test descriptor index field is set to a non-zero value, then the SELF-TEST CODE field is unspecified (see 3.4.2). If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SELF-TEST CODE field is unspecified (see 3.4.2).

Table 87 — Self-Test Results log parameters (part 2 of 4)

Field	Description or reference
SELF-TEST RESULTS	The SATL shall read the ATA log data as defined in 10.2.5.2. If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the SELF-TEST RESULTS field to zero for each log parameter returned. If the value contained in the Self-test descriptor index field is set to a nonzero value, then the SATL shall set the SELF-TEST RESULTS field to: a) the value contained in the Self-test Execution Status bits of the content of the self-test execution status byte field of the n^{th} descriptor entry, where n is equal to the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is greater than zero (e.g., for a log parameter with the PARAMETER CODE field of 0002h and a value contained in the Selftest descriptor index field of 6h, then the fifth descriptor entry is used); or b) zero, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is less than or equal to zero. If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SATL shall set the SELF-TEST RESULTS field to the value contained in the Self-test Execution Status bits of the content of the self-test execution status byte field of the nth descriptor entry, where n is equal to the value contained in the PARAMETER CODE field for the log parameter being returned (e.g., for a log parameter with the PARAMETER CODE field of 0002h, then the second descriptor entry is used).
SELF-TEST NUMBER	Unspecified (see 3.4.2)

Table 87 — Self-Test Results log parameters (part 3 of 4)

Field	Description or reference
ACCUMULATED POWER ON HOURS	The SATL shall read the ATA log data as defined in 10.2.5.2. If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the TIMESTAMP field to zero for each log parameter returned. If the value contained in the Self-test descriptor index field is set to a nonzero value, then the SATL shall set the TIMESTAMP field to: a) the values contained in the Life timestamp (most significant byte) field and Life timestamp (least significant byte) field of the n^{th} descriptor entry, where n is equal to the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one, if the result of value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is greater than zero (e.g., for a log parameter with the PARAMETER CODE field of 0002h and a value contained in the Selftest descriptor index field of 6h, then the fourth descriptor entry is used); or b) zero, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is less than or equal to zero. If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SATL shall set the TIMESTAMP field to the values contained in the Life timestamp (most significant byte) field and Life timestamp (least significant byte) field of the n^{th} descriptor entry, where n is equal to the value contained in the PARAMETER CODE field for the log parameter being returned (e.g., for a log parameter with the PARAMETER CODE field of 0002h, then the second descriptor entry is used).

Table 87 — Self-Test Results log parameters (part 4 of 4)

Field	Description or reference
ADDRESS OF FIRST FAILURE	The SATL shall read the ATA log data as defined in 10.2.5.2. If the SATL reads the ATA log data using the ATA READ LOG EXT command specifying the Extended SMART self-test log, then the SATL shall check if the value contained in the Self-test descriptor index field in the first block of data (i.e., bytes 2 and 3) is set to zero. If the value contained in the Self-test descriptor index field is set to zero, then the SATL shall set the ADDRESS OF FIRST FAILURE field to zero for each log parameter returned. If the value contained in the Self-test descriptor index field is set to a nonzero value, then the SATL shall set the ADDRESS OF FIRST FAILURE field to: a) the values contained in the Failing LBA (47:40) field, Failing LBA (39:32) field, Failing LBA (31:24) field, Failing LBA (23:16) field, Failing LBA (15:8) field, and Failing LBA (7:0) field of the n^{th} descriptor entry, where n is equal to the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is greater than zero (e.g., for a log parameter with the PARAMETER CODE field of 0002h and a value contained in the Self-test descriptor index field of 6h, then the fourth descriptor entry is used); or b) zero, if the result of the value contained in the Self-test descriptor index field minus the value contained in the PARAMETER CODE field for the log parameter being returned plus one is less than or equal to zero. If the SATL reads the ATA log data using the ATA SMART READ LOG command specifying the SMART self-test log, then the SATL shall set the ADDRESS OF FIRST FAILURE field to the values contained in the Failing LBA (27:24) field, Failing LBA (23:16) field, Failing LBA (15:8) field, and Failing LBA (7:0) field of the n^{th} descriptor entry, where n is equal to the value contained in the PARAMETER CODE field for the log parameter being returned (e.g., for a log parameter with the PARAMETER CODE field of 0002h, then the second descriptor entry is used).
SENSE KEY	10.2.5.3
ADDITIONAL SENSE CODE	10.2.5.3
ADDITIONAL SENSE CODE QUALIFIER	10.2.5.3

10.2.5.2 A method of determining ATA command selection for field translations

To translate the SELF-TEST CODE field, the SELF-TEST RESULTS field, the TIMESTAMP field, the ADDRESS OF FIRST FAILURE field, the SENSE KEY field, the ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field of Self-Test Results log parameters, the SATL shall send an ATA IDENTIFY DEVICE command to the ATA device, and from the returned data the SATL shall determine if the ATA device supports the 48-bit Address feature set. If the 48-bit Address feature set is supported (i.e., bit 10 of word 83 of ATA IDENTIFY DEVICE data is set to one), then the SATL shall send an ATA READ LOG EXT command with the Log address set to 07h (i.e., Extended SMART self-test log) to the ATA device. If the 48-bit Address feature set is not supported (i.e., bit 10 of word 83 of ATA IDENTIFY DEVICE data is set to zero), then the SATL shall send

an ATA SMART READ LOG command with the Log address set to 06h (i.e., SMART self-test log) to the ATA device.

10.2.5.3 Sense key and additional sense code

The SATL shall determine the SENSE KEY field, the ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field returned in each log parameter from the content of the self-test execution status byte returned from a ATA READ LOG EXT command or ATA SMART READ LOG command sent to the ATA device (see 10.2.5.2). The values returned in each log parameter shall be translated into sense data for the sense key, and additional sense code as shown in table 88.

Table 88 — ATA Self-test execution status values translated to SCSI sense keys and sense codes

ATA	SCSI		
Self-Test execution status value	**Sense key**	**Additional sense code**	**NN**
0	NO SENSE	NO ADDITIONAL SENSE INFORMATION	n/a
1	ABORTED COMMAND	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	81h
2		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	82h
3		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	83h
4	HARDWARE ERROR	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	84h
5		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	85h
6		DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	86h
7	MEDIUM ERROR	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	87h
8	HARDWARE ERROR	DIAGNOSTIC FAILURE ON COMPONENT NN (80h - FFh)	88h
9-14	Unspecified (see 3.4.2)[a]		
15	NO SENSE	NO ADDITIONAL SENSE INFORMATION	n/a

[a] Self-Test execution status values from 9 to 14 are reserved in ATA8-ACS.

10.2.6 Informational Exceptions log page

10.2.6.1 Informational Exceptions log page overview

The Informational Exceptions log page provides detail about informational exceptions. Table 89 shows the log page header fields.

Table 89 — Informational Exceptions log page header fields

Field	Description or reference
PAGE CODE	Shall be set to 2Fh. The SATL shall send the ATA SMART RETURN STATUS command to the ATA device. Data returned from the ATA device shall be translated into the appropriate log sense parameter data (see 10.2.6.2) to be returned to the application client.
SPF	Shall be set to zero
DS	Unspecified (see 3.4.2)
SUBPAGE CODE	Shall be set to zero
PAGE LENGTH	Unspecified (see 3.4.2)

The first log parameter is the informational exceptions general parameter shown in table 90.

Table 90 — Informational Exceptions general parameter data

Field	Description or reference
PARAMETER CODE	Shall be set to 0000h
DU	Shall be set to zero
TSD	Shall be set to zero
ETC	Shall be set to zero
TMC	Shall be set to zero
LINKING AND FORMATTING	Shall be set to 11b
PARAMETER LENGTH	Unspecified (see 3.4.2)
INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE	10.2.6.2
INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER	10.2.6.2
MOST RECENT TEMPERATURE READING	10.2.6.3
Vendor specific	Unspecified (see 3.4.2)

10.2.6.2 Additional sense code and additional sense code qualifier translations

Data received from a ATA device in response to an ATA SMART RETURN STATUS command shall be translated by the SATL into the informational exceptions general parameter data returned to the application

client. Table 91 provides the parameter data translations for the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE field and the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER field.

Table 91 — ATA SMART RETURN STATUS translations

Data returned to SATL from the ATA device by the ATA SMART RETURN STATUS command	SMART condition	Informational exceptions general parameter data fields
LBA Mid = 4Fh LBA High = C2h	threshold not exceeded	INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE = 00h, INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER = 00h
LBA Mid = F4h LBA High = 2Ch	threshold exceeded	INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE = 5Dh, INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER = 10h

10.2.6.3 Most recent temperature reading translation

If the ATA device supports the SCT Feature Set (see ATA8-ACS), then to translate the MOST RECENT TEMPERATURE READING field of the Informational Exceptions log page, the SATL shall send an ATA SCT Status Request to the ATA device; and then:

a) if the HDA Temp field (see SCT) is less than zero, the SATL shall set the MOST RECENT TEMPERATURE READING field to zero;
b) if the HDA Temp field is equal to 80h, the SATL shall set the MOST RECENT TEMPERATURE READING field to FFh; or
c) the SATL shall set the MOST RECENT TEMPERATURE READING FIELD to the value in the HDA Temp field.

If the ATA device does not support the SCT feature set, then the SATL shall set the MOST RECENT TEMPERATURE READING field to FFh.

10.3 Vital product data parameters

10.3.1 Vital product data parameters overview

Table 92 provides a summary of the VPD page translations defined in this standard.

Table 92 — Summary of SCSI / ATA VPD page mapping

SCSI VPD page	Reference
Supported VPD Pages VPD page (i.e., 00h)	10.3.2
Unit Serial Number VPD page (i.e., 80h)	10.3.3
Device Identification VPD page (i.e., 83h)	10.3.4
Mode Page Policy VPD page (i.e., 87h)	10.3.5
ATA Information VPD page (i.e., 89h)	12.4.2
Block Device Characteristics VPD Page (i.e., B1h)	10.3.6
All others	See SPC-4 and SBC-3 Unspecified (see 3.4.2)

10.3.2 Supported VPD pages VPD page

Table 93 shows the fields of the Supported VPD pages VPD page.

Table 93 — Supported VPD pages VPD page fields

Field	Description or reference
PERIPHERAL QUALIFIER	The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.
PERIPHERAL DEVICE TYPE	
PAGE CODE	The SATL shall set this field to 00h.
PAGE LENGTH	The SATL shall set this field to indicate the length of the supported VPD page list returned in number of bytes.
Supported VPD page list	Unspecified (see 3.4.2).

10.3.3 Unit Serial Number VPD page

Table 94 shows the fields of the Unit Serial Number VPD page.

Table 94 — Unit Serial Number VPD page fields

Field	Description or reference
PERIPHERAL QUALIFIER	The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.
PERIPHERAL DEVICE TYPE	
PAGE CODE	The SATL shall set this field to 80h.
PAGE LENGTH	Shall be set to 14h
PRODUCT SERIAL NUMBER	The PRODUCT SERIAL NUMBER field contains a representation of the Serial number field in the ATA IDENTIFY DEVICE data words 19:10 last retrieved from the ATA device. Each pair of bytes in the Serial number field shall be swapped to create a valid ASCII string format in the PRODUCT SERIAL NUMBER field as described in table 95

Table 95 shows the positional swapping of ATA IDENTIFY DEVICE data fields to bytes in the PRODUCT SERIAL NUMBER field.

Table 95 — PRODUCT SERIAL NUMBER field

Byte	Contents
0	IDENTIFY DEVICE word 10 bits 15:8 (i.e., byte 1)
1	IDENTIFY DEVICE word 10 bits 7:0 (i.e., byte 0)
2	IDENTIFY DEVICE word 11 bits 15:8 (i.e., byte 3)
3	IDENTIFY DEVICE word 11 bits 7:0 (i.e., byte 2)
...	...
18	IDENTIFY DEVICE word 19 bits 15:8 (i.e., byte 19)
19	IDENTIFY DEVICE word 19 bits 7:0 (i.e., byte 18)

NOTE 8 - Although SPC-4 defines the PRODUCT SERIAL NUMBER field as right-aligned, ATA8-ACS does not require its SERIAL NUMBER field to be right-aligned. Therefore, the PRODUCT SERIAL NUMBER field for SAT may not be right-aligned.

10.3.4 Device Identification VPD page

10.3.4.1 Device Identification VPD page overview

The SATL shall return the Device Identification VPD page (see SPC-4) as defined in table 96.

Table 96 — Device Identification VPD page fields

Field	Description or reference
PERIPHERAL QUALIFIER	The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.
PERIPHERAL DEVICE TYPE	
PAGE CODE	The SATL shall set this field to 83h.
PAGE LENGTH	Shall be set to the length of the remaining bytes of the VPD page.
Designation descriptor	One designation descriptor for a logical unit (i.e., a logical unit name) shall be included (see 10.3.4.2). In some environments, one or more additional designation descriptors may be included (see 10.3.4.3).

10.3.4.2 Logical unit name

10.3.4.2.1 Logical unit name overview

If the ATA device returns the ATA IDENTIFY DEVICE data word 87 bit 8 is set to one indicating that the ATA device supports the World wide name field (i.e., ATA IDENTIFY DEVICE data words 111:108), then the SATL shall include an designation descriptor containing a logical unit name as defined in 10.3.4.2.2.

If the ATA IDENTIFY DEVICE data returned by the ATA device word 87 bit 8 is set to zero, indicating that the ATA device does not support the World wide name field (i.e., ATA IDENTIFY DEVICE data words 111:108), then the SATL shall include an identification descriptor containing a logical unit name as defined in 10.3.4.2.3.

10.3.4.2.2 Logical unit name derived from the world wide name

Table 97 defines the logical unit name derived from the ATA device world wide name.

Table 97 — Logical unit name derived from the world wide name

Field	Description or reference
PROTOCOL IDENTIFIER	Shall be set to zero
CODE SET	Shall be set to 1h
PIV	Shall be set to zero
ASSOCIATION	Shall be set to zero
DESIGNATOR TYPE	Shall be set to 3h
DESIGNATOR LENGTH	Shall be set to 08h
NAA	See table 98
IEEE COMPANY_ID	See table 98
VENDOR SPECIFIC IDENTIFIER	See table 98

The NAA field, the IEEE COMPANY_ID field, and the VENDOR SPECIFIC IDENTIFIER field shall be based on the ATA IDENTIFY DEVICE data World wide name field as described in table 98.

Table 98 — Fields in the logical unit name

Field		Contents
Field name	**Specific bits in table 97**	
NAA	Byte 4 bits 7:4	IDENTIFY DEVICE word 108 bits 15:12[a]
IEEE COMPANY_ID	Byte 4 bits 3:0	IDENTIFY DEVICE word 108 bits 11:8
	Byte 5	IDENTIFY DEVICE word 108 bits 7:0
	Byte 6	IDENTIFY DEVICE word 109 bits 15:8
	Byte 7 bits 7:4	IDENTIFY DEVICE word 109 bits 7:4
VENDOR SPECIFIC IDENTIFIER	Byte 7 bits 3:0	IDENTIFY DEVICE word 109 bits 3:0
	Byte 8	IDENTIFY DEVICE word 110 bits 15:8
	Byte 9	IDENTIFY DEVICE word 110 bits 7:0
	Byte 10	IDENTIFY DEVICE word 111 bits 15:8
	Byte 11	IDENTIFY DEVICE word 111 bits 7:0

[a] This 4-bit field is required to be set to 5h (i.e., IEEE Registered) by ATA8-ACS.

10.3.4.2.3 Logical unit name derived from the model number and serial number

Table 99 defines the logical unit name derived from the ATA device model number and serial number.

Table 99 — Logical unit name derived from the world wide name

Field	Description or reference
PROTOCOL IDENTIFIER	Shall be set to zero
CODE SET	Shall be set to 2h
PIV	Shall be set to zero
ASSOCIATION	Shall be set to zero
DESIGNATOR TYPE	Shall be set to 1h
DESIGNATOR LENGTH	Shall be set to 68
T10 VENDOR IDENTIFICATION	Shall be set to the string ‘ATA¬¬¬¬¬¬’.
VENDOR SPECIFIC IDENTIFIER	See table 100

The VENDOR SPECIFIC IDENTIFIER field shall be set to a representation of the ATA IDENTIFY DEVICE data Model number field concatenated with a representation of the ATA IDENTIFY DEVICE data Serial number field as described in table 100.

Table 100 — VENDOR SPECIFIC IDENTIFIER field for logical unit name

Byte	Contents	
	Source field name	Source location
0	Model number field	IDENTIFY DEVICE word 27 bits 15:8
1		IDENTIFY DEVICE word 27 bits 7:0
2		IDENTIFY DEVICE word 28 bits 15:8
...		...
39		IDENTIFY DEVICE word 46 bits 7:0
40	Serial number field	IDENTIFY DEVICE word 10 bits 15:8
41		IDENTIFY DEVICE word 10 bits 7:0
42		IDENTIFY DEVICE word 11 bits 15:8
...		...
59		IDENTIFY DEVICE word 19 bits 7:0

NOTE 9 - The logical unit name using the T10 vendor ID based format is not guaranteed to be worldwide unique, since ATA8-ACS only requires the combination of the Model number field and Serial number field to be unique for a given manufacturer but defines no manufacturer identification field.

10.3.4.3 Examples of additional designation descriptors

10.3.4.3.1 Designation descriptors included by a SATL in an ATA host

Figure 12 shows the designation descriptor returned by a SATL in an ATA host (i.e., where the ATA device is being accessed with an ATA host port) containing a logical unit name based on ATA IDENTIFY DEVICE data (see table 97 or table 99 in 10.3.4.2).

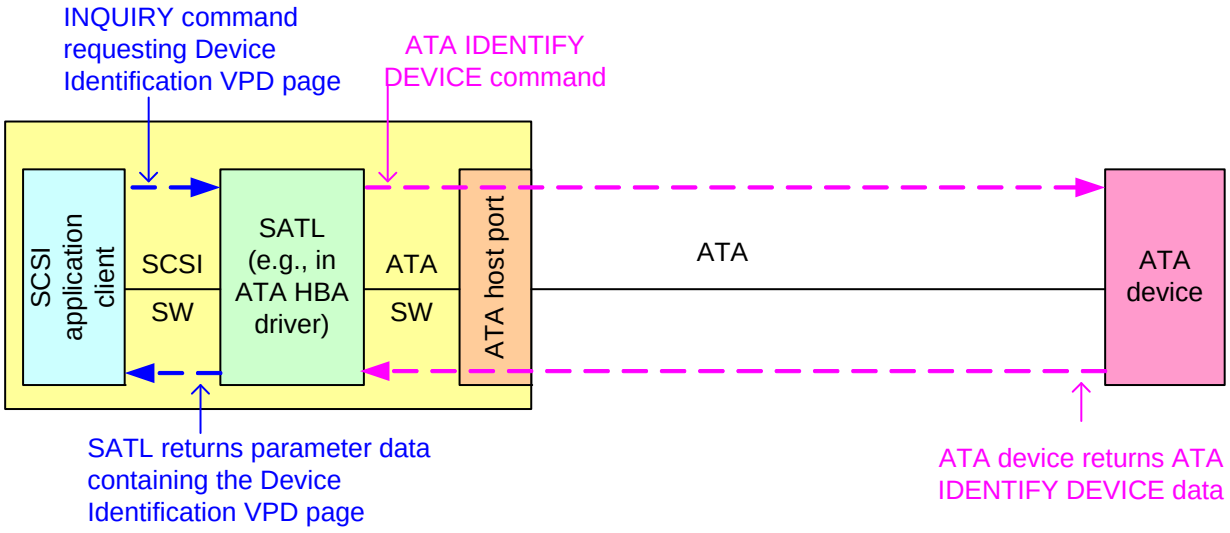


Figure 12 — Designation descriptors included by a SATL in an ATA host

10.3.4.3.2 Designation descriptors included by a SATL in a SAS initiator device

Figure 13 shows the designation descriptors returned by a SATL in a SAS initiator device (i.e., where the ATA device is being accessed by a SAS STP initiator port through an STP/SATA bridge) that contain:

a) a logical unit name based on ATA IDENTIFY DEVICE data (see table 97 or table 99 in 10.3.4.2);
b) a target port identifier based on the SAS STP target port SAS address (see table 101); and
c) a relative target port identifier set to 0001h (see SPC-4).

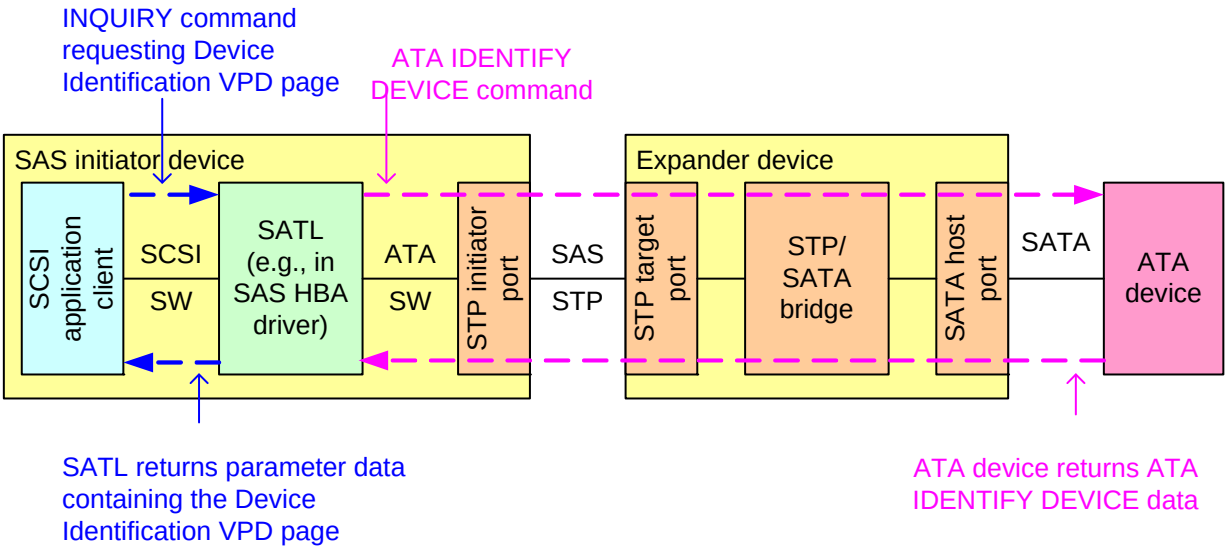


Figure 13 — Designation descriptors included by a SATL in a SAS initiator device

The SATL includes a target port identifier as defined in table 101.

Table 101 — Target port identifier for SAS

Byte\Bit	7	6	5	4	3	2	1	0
0	PROTOCOL IDENTIFIER (6h)				CODE SET (1h)			
1	PIV (1b)	Reserved		ASSOCIATION (01b)		DESIGNATOR TYPE (3h)		
2	Reserved							
3	DESIGNATOR LENGTH (08h)							
4	SAS ADDRESS							
11								

The CODE SET field is set to 1h (i.e., binary).

The PIV bit is set to one.

The ASSOCIATION field is set to 01b (i.e., target port).

The DESIGNATOR TYPE field is set to 3h (i.e., NAA).

The SAS ADDRESS field is set to the SAS address of the STP target port providing the STP/SATA bridge functionality (i.e., the SAS address of the SATA device).

10.3.4.3.3 Designation descriptors included by a SATL in a SCSI to ATA protocol bridge

Figure 14 shows the designation descriptors returned by a SATL in a SCSI to ATA protocol bridge, where the ATA device is being accessed by an ATA host port, and the SATL is being accessed with a SCSI target port using a SCSI transport protocol (e.g, FCP-3 or iSCSI) that contains:

a) a logical unit name based on ATA IDENTIFY DEVICE data (see table 97 or table 99 in 10.3.4.2);
b) any target port identifiers specified by the SCSI transport protocol standard; and
c) any other designation descriptors supported by the protocol bridge (e.g., a target device name).

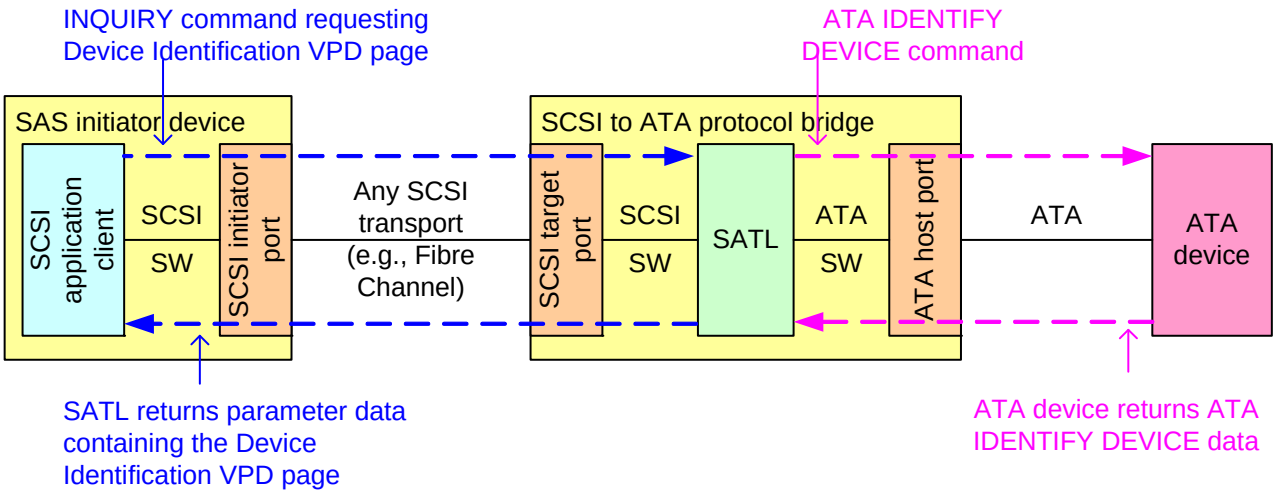


Figure 14 — Designation descriptors included by a SATL in a SCSI to ATA protocol bridge

10.3.5 Mode Page Policy VPD page

The SATL should implement the Mode Page Policy VPD page for each logical unit emulated (see SPC-4). Table 102 defines the Mode Page Policy VPD page (see SPC-4) returned by the SATL.

Table 102 — Mode Page Policy VPD page fields

Field	Description or reference
PERIPHERAL QUALIFIER	The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.
PERIPHERAL DEVICE TYPE	
PAGE CODE	The SATL shall set this field to 87h.
PAGE LENGTH	Unspecified (see 3.4.2).
Mode page policy descriptor	If the SATL implements the Mode Page Policy VPD page, then the SATL shall include at least one mode page policy descriptor (see table 103).

Table 103 shows the fields of the mode page policy descriptor. See 10.1.1 for recommendations on implementation of the fields in table 103.

Table 103 — Mode policy descriptor for SAT

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved		POLICY PAGE CODE					
1	POLICY SUBPAGE CODE							
2	MLUS	Reserved						MODE PAGE POLICY
3	Reserved							

The POLICY PAGE CODE field, the POLICY SUBPAGE CODE field, the multiple logical units share (i.e., MLUS) bit, and MODE PAGE POLICY field are unspecified (see 3.4.2 and SPC-4).

10.3.6 Block Device Characteristics VPD Page

Table 104 shows the translation of fields in the Block Device Characteristics VPD page.

Table 104 — Block Device Characteristics VPD Page field translations

Field	Description or Reference
PERIPHERAL DEVICE TYPE	The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.
PERIPHERAL QUALIFIER	
PAGE CODE	Shall be set to B1h.
PAGE LENGTH	Shall be set to 3Ch.
MEDIUM ROTATION RATE	The SATL shall set this field to the value contained in the ATA IDENTIFY DEVICE data word 217.
NOMINAL FORM FACTOR	The SATL shall set this field to the value contained in the ATA IDENTIFY DEVICE data word 168 bits 3:0.

11 Translation of ATA errors to SCSI errors

ATA device errors are translated to the appropriate SCSI errors. The ATA Status register and Error register bit settings provide the information to be translated into SCSI sense key, additional sense code, and additional sense code qualifier for error reporting. Unless otherwise specified in the subclause describing the translation of a particular SCSI command, log page, mode page or VPD page, the SATL shall translate ATA device errors to SCSI errors as shown in table 105.

Table 105 — Translation of ATA errors to SCSI errors

ATA Error Register		SCSI Error	
Status	**Error**[a]	**Sense key**	**Additional sense code**
DF[b]	n/a	HARDWARE ERROR	INTERNAL TARGET FAILURE
ERR	NM	NOT READY	MEDIUM NOT PRESENT
ERR	UNC	MEDIUM ERROR	UNRECOVERED READ ERROR
ERR	WP	DATA PROTECT	WRITE PROTECTED
ERR	IDNF	ILLEGAL REQUEST[d]	LOGICAL BLOCK ADDRESS OUT OF RANGE[d]
ERR	ABRT[c]	ABORTED COMMAND	NO ADDITIONAL SENSE INFORMATION
ERR	MC	UNIT ATTENTION	NOT READY TO READY CHANGE, MEDIUM MAY HAVE CHANGED
ERR	MCR	UNIT ATTENTION	OPERATOR MEDIUM REMOVAL REQUEST
ERR	ICRC	ABORTED COMMAND	INFORMATION UNIT iuCRC ERROR DETECTED
CORR	n/a	This condition is not considered an error.	

[a] If the Error register has an obsolete bit set to one, then the SATL may return a vendor-specific additional sense code (e.g., if the AMNF bit is set to one, return MEDIUM ERROR - ADDRESS MARK NOT FOUND FOR DATA FIELD).

[b] After an ATA device returns a DF bit set to one, the SATL processes any subsequent commands received for the emulated logical unit corresponding to the ATA device by terminating the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to INTERNAL TARGET FAILURE.

[c] The ABRT bit is ignored if any other ATA error bit is set.

[d] SATLs compliant with previous versions of this standard return a sense key of MEDIUM ERROR and an additional sense code of RECORD NOT FOUND.

12 SAT-specific SCSI extensions

12.1 SAT-specific SCSI extensions overview

This clause defines additional SCSI commands, mode pages, security protocols, and VPD pages that may be supported by a SATL to provide capabilities beyond those defined in the other SCSI command sets.

SCSI commands defined for SATL implementations include:

a) ATA PASS-THROUGH (12) command (see 12.2.2); and
b) ATA PASS-THROUGH (16) command (see 12.2.3).

Mode pages defined for SATL implementations include:

a) PATA Control mode page (see 12.3.2); and
b) ATA Power Condition mode page (see 12.3.3).

VPD pages defined for SATL implementations include:

a) ATA Information VPD page (see 12.4.2).

12.2 ATA PASS-THROUGH commands

12.2.1 ATA PASS-THROUGH commands overview

ATA PASS-THROUGH commands provide a method for:

a) an application client to transmit an ATA command to an ATA device;
b) optionally, transferring data between an application client and an ATA device; and
c) an ATA device to transfer completion status through the SATL.

This is accomplished by defining:

a) CDBs containing ATA command information (see 12.2.2 and 12.2.3); and
b) specific SCSI status and sense data usage for returning the results of an ATA command (see 12.2.5 and 12.2.7).

If the SATL supports the ATA PASS-THROUGH (12) command or the ATA PASS-THROUGH (16) command, then the SATL shall support the ATA Status Return descriptor (see 12.2.6).

12.2.2 ATA PASS-THROUGH (12) command

Table 106 shows the CDB for the ATA PASS-THROUGH (12) command.

Table 106 — ATA PASS-THROUGH (12) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (A1h)							
1	MULTIPLE_COUNT			PROTOCOL				Reserved
2	OFF_LINE		CK_COND	Reserved	T_DIR	BYTE_BLOCK	T_LENGTH	
3	FEATURES (7:0)							
4	SECTOR_COUNT (7:0)							
5	LBA_LOW (7:0)							
6	LBA_MID (7:0)							
7	LBA_HIGH (7:0)							
8	DEVICE							
9	COMMAND							
10	Reserved							
11	CONTROL (see 6.5)							

12.2.4 describes the mapping between the fields in the ATA PASS-THROUGH (12) CDB to corresponding ATA command fields (see ATA8-ACS).

If the SATL receives an ATA PASS-THROUGH (12) command, then the SATL shall check the PROTOCOL field (see table 107) to determine the type of action requested.

Table 107 — PROTOCOL field

Code	Description
0h	Device Management - ATA hardware reset
1h	Device Management - ATA software reset
2h	Reserved
3h	Non-data
4h	PIO Data-In
5h	PIO Data-Out
6h	DMA
7h	DMA Queued
8h	Execute Device Diagnostic
9h	Non-data command - Device Reset
Ah	UDMA Data In
Bh	UDMA Data Out
Ch	FPDMA[a]
Dh, Eh	Reserved
Fh	Return Response Information

[a] See SATA-2.6.

The PROTOCOL field specifies the protocol to use when the ATA device processes the command. ATA8-AAM defines the meaning of protocol values ranging from 0h to Bh.

If the PROTOCOL field specified is in the range from 3h to Ch, then the SATL shall send an ATA command specified by the CDB to the ATA device.

If the PROTOCOL field contains Fh(i.e., Return Response Information), then the SATL shall:

1) ignore all fields in the CDB except for the PROTOCOL field;
2) read the ATA Command Block as follows:
   A) if the transport is SATA, read the current Shadow Command Block registers; or
   B) if the transport is PATA, read the current Command Block registers; and
3) return the contents of the ATA Command Block in the ATA Status Return Descriptor as defined in 12.2.6.

If the value in the PROTOCOL field is inappropriate for the command specified in the COMMAND field (see ATA8-ACS), then the SATL may lose communication with the ATA device. This standard does not specify the SATL behavior if this occurs.

If the PROTOCOL field is set to 0h (i.e., ATA Hardware Reset) and the device is a PATA device, then the SATL shall assert RST- (see ATA8-APT). If the PROTOCOL field is set to zero (i.e., ATA Hardware Reset) and the device is a SATA device, then the SATL shall send a COMRESET to the SATA device. When this protocol is selected, only the PROTOCOL field and the OFF_LINE field are valid. The SATL shall ignore all other fields in the CDB.

If the PROTOCOL field is set to one, then the SATL shall send a software reset to the ATA device (see ATA8-AAM). When this protocol is selected, only the PROTOCOL field and the OFF_LINE field are valid. The SATL shall ignore all other fields in the CDB.

If the value in the PROTOCOL field requests the SATL to send a command to the ATA device, then the SATL shall set the fields in the ATA command using fields in the ATA PASS-THROUGH CDB as shown in table 111.

The SATL shall determine if a data transfer is necessary and how to perform the data transfer by examining values in the MULTIPLE_COUNT field, PROTOCOL field, OFF_LINE field, T_DIR bit, BYTE_BLOCK bit, and T_LENGTH field. The SATL shall ignore the COMMAND field in the CDB except to copy the COMMAND field in the CDB to the Command field in the Register – Host to Device FIS or to the ATA Command register. If the ATA command completes with an error, then the SATL shall return the Error Output fields (see ATA8-ACS) in the sense data.

The SATL shall configure the ATA host and the ATA device for the PIO, DMA, and UDMA transfer rates that both the SATL and ATA device support. The SATL should set the transfer rates to the maximum supported by both the SATL and the ATA device. The ATA PASS-THROUGH (12) command should not be used to send an ATA SET FEATURES command that changes the PIO/DMA/UDMA or other transfer modes of the ATA device. The result of an ATA SET FEATURES command that changes the PIO/DMA/UDMA or other transfer modes of the ATA device is outside the scope of this standard and may cause communication to be lost with the ATA device; thus preventing the SATL from performing any action based on the contents of the CDB.

The BYTE_BLOCK (Byte/Block) bit specifies whether the transfer length in the location specified by the T_LENGTH field specifies the number of bytes to transfer or the number of blocks to transfer. If the value in the BYTE_BLOCK bit is set to zero, then the SATL shall transfer the number of bytes specified in the location specified by the T_LENGTH field. If the value in the BYTE_BLOCK bit is set to one then the SATL shall transfer the number of blocks specified in the location specified by the T_LENGTH field. The SATL shall ignore the BYTE_BLOCK bit when the T_LENGTH field is set to zero.

The CK_COND (Check Condition) bit may be used to request the SATL to return a copy of ATA register information in the sense data upon command completion. If the CK_COND bit is set to one then the SATL shall return a status of CHECK CONDITION when the ATA command completes, even if the command completes successfully, and return the ATA Normal Output fields (see ATA8-ACS) in the sense data. If the CK_COND bit is set to zero, then the SATL shall terminate the command with CHECK CONDITION status only if an error occurs in processing the command. See clause 11 for a description of ATA error conditions.

The DEVICE field specifies a value for the SATL to load into the ATA Device field. Table 108 shows the bits in the DEVICE field.

Table 108 — ATA PASS-THROUGH (12) command and ATA PASS-THROUGH (16) command DEVICE field

Bit							
7	6	5	4	3	2	1	0
Obsolete	Command Specific	Obsolete	DEV	Command Specific			

The SATL shall ignore the DEV bit in the DEVICE field of the CDB.

The SATL shall set the value of the DEV bit in the ATA device register based upon the SATL mapping of ATA devices to I_T_L nexuses.

If the PROTOCOL field specifies a PIO data transfer, then the SATL shall perform a PIO type transfer. The MULTIPLE_COUNT field specifies the logarithm base 2 of the number of logical sectors an ATA host shall transfer per DRQ Data Block (e.g, if the field is set to 4, the SATL shall transfer 2^4 (i.e., 16) logical sectors of data in each DRQ Data Block). If the MULTIPLE_COUNT field is nonzero and the COMMAND field is not an ATA READ MULTIPLE command, ATA READ MULTIPLE EXT command, ATA WRITE MULTIPLE command, ATA WRITE MULTIPLE EXT command, or ATA WRITE MULTIPLE FUA EXT command, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The OFF_LINE field specifies the time period during which the ATA Status register and the ATA Alternate Status register may be invalid after command acceptance. In a SATL with a PATA device attached, some commands may cause the PATA device to place the ATA bus in an indeterminate state. This may cause the ATA host to see command completion before the command is completed. When the application client sends a command that is capable of placing the bus in an indeterminate state, it shall set the OFF_LINE field to a value that specifies the maximum number of seconds from the time a command is sent until the ATA Status register is valid. The SATL shall not use the ATA Status register or ATA Alternate Status register to determine ATA command completion status until this time has elapsed. The valid status is available (2^{off_line+1} - 2) seconds (i.e., 0, 2, 6, and 14 seconds) after the command register is stored.

NOTE 10 - If the application client specifies an off_line value that is too small, then the results are indeterminate and may compromise the integrity of the data.

If the Transfer Direction (T_DIR) bit and the direction of the data transfer specified in the PROTOCOL field do not match, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the T_DIR bit is set to zero, then the SATL shall transfer data from the application client to the ATA device. If the T_DIR bit is set to one, then the SATL shall transfer data from the ATA device to the application client. The SATL shall ignore the T_DIR bit if the T_LENGTH field is set to zero.

The Transfer Length (T_LENGTH) field specifies where in the CDB the SATL shall locate the transfer length for the command (see table 109).

Table 109 — T_LENGTH field

Code	Description
00b	No data is transferred
01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field.
10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field.
11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.94).

See 12.2.4 for a description of the mapping from the FEATURES (7:0) field, the SECTOR_COUNT (7:0) field, the LBA_LOW (7:0) field, the LBA_MID (7:0) field, the LBA_HIGH (7:0) field, the DEVICE field, and the COMMAND field in the ATA PASS-THROUGH (12) CDB to corresponding ATA command fields (see ATA8-ACS).

12.2.3 ATA PASS-THROUGH (16) command

Table 110 shows the format of the ATA PASS-THROUGH (16) command.

Table 110 — ATA PASS-THROUGH (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (85h)							
1	MULTIPLE_COUNT			PROTOCOL				EXTEND
2	OFF_LINE		CK_COND	Reserved	T_DIR	BYTE_BLOCK	T_LENGTH	
3	FEATURES (15:8)							
4	FEATURES (7:0)							
5	SECTOR_COUNT (15:8)							
6	SECTOR_COUNT (7:0)							
7	LBA_LOW (15:8)							
8	LBA_LOW (7:0)							
9	LBA_MID (15:8)							
10	LBA_MID (7:0)							
11	LBA_HIGH (15:8)							
12	LBA_HIGH (7:0)							
13	DEVICE							
14	COMMAND							
15	CONTROL (see 6.5)							

If the EXTEND bit is set to zero, then the FEATURES (15:8) field, the SECTOR_COUNT (15:8) field, the LBA_LOW (15:8) field, then the LBA_MID (15:8) field, and the LBA_HIGH (15:8) field shall be ignored by the SATL, and the SATL shall process this command as specified in 12.2.2.

If the EXTEND bit is set to one, then the FEATURES (15:8) field, the SECTOR_COUNT (15:8) field, the LBA_LOW (15:8) field, then the LBA_MID (15:8) field, and the LBA_HIGH (15:8) field are valid, and the SATL shall process this command as specified in 12.2.2 except as described in the remainder of this subclause

If the EXTEND bit is set to one and the value in the PROTOCOL field requests the SATL to send an ATA command to the device, then the SATL shall send a 48 bit ATA command to the ATA device.

See 12.2.2 for a description of the MULTIPLE_COUNT field, the PROTOCOL field, the OFF_LINE field, the CK_COND bit, the T_DIR bit, and the BYTE_BLOCK bit.

See 12.2.4 for a description of the mapping from the FEATURES (15:8) field, the FEATURES (7:0) field, the SECTOR_COUNT (15:8) field, the SECTOR_COUNT (7:0) field, the LBA_LOW (15:8) field, the LBA_LOW (7:0) field, the LBA_MID (15:8) field, the LBA_MID (7:0) field, the LBA_HIGH (15:8) field, the LBA_HIGH (7:0) field, the DEVICE field, and the COMMAND field in the ATA PASS-THROUGH (16) CDB to corresponding ATA command fields (see ATA8-ACS).

12.2.4 Mapping of ATA PASS-THROUGH CDB field translations

Table 111 shows the mapping between the fields in the ATA PASS-THROUGH (12) CDB and the ATA PASS-THROUGH (16) CDB to corresponding ATA command fields (see ATA8-ACS).

Table 111 — Mapping of ATA PASS-THROUGH (16) CDB fields to ATA command fields

CDB field	48-bit ATA command field[a]	28-bit ATA command field[b]
FEATURES (15:8)	Features (15:8)	n/a
FEATURES (7:0)	Features (7:0)	Features (7:0)
SECTOR_COUNT (15:8)	Count (15:8)	n/a
SECTOR_COUNT (7:0)	Count (7:0)	Count (7:0)
LBA_LOW (15:8)	LBA (31:24)	n/a
LBA_LOW (7:0)	LBA (7:0)	LBA (7:0)
LBA_MID (15:8)	LBA (39:32)	n/a
LBA_MID (7:0)	LBA (15:8)	LBA (15:8)
LBA_HIGH (15:8)	LBA (47:40)	n/a
LBA_HIGH (7:0)	LBA (23:16)	LBA (23:16)
DEVICE (7:4)	Device (7:4)	Device (7:4)
DEVICE (3:0)	Device (3:0)	LBA (27:24)
COMMAND	Command	Command

[a] The 48-bit ATA command translation applies only to the ATA PASS-THROUGH (16) command, and not to the ATA PASS-THROUGH (12) command.

[b] The 28-bit ATA command translation may apply to either the ATA PASS-THROUGH (12) command or the ATA PASS-THROUGH (16) command.

The SATL shall determine the transfer length by the method specified in the T_LENGTH field as shown in table 112.

Table 112 — EXTEND bit and T_LENGTH field

EXTEND	T_LENGTH	Description
0	00b	No data is transferred.
	01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field.
	10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field.
	11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.94).
1	00b	No data is transferred.
	01b	The transfer length is an unsigned integer specified in the FEATURES (7:0) field and the FEATURES (15:8) field.
	10b	The transfer length is an unsigned integer specified in the SECTOR_COUNT (7:0) field and the SECTOR_COUNT (15:8) field.
	11b	The transfer length is an unsigned integer specified in the TPSIU (see 3.1.94) STPSIU field.

12.2.5 ATA PASS-THROUGH status return

Table 113 shows the possible results of ATA PASS-THROUGH (12) command or ATA PASS-THROUGH (16) command processing depending on the value of the CK_COND bit in the CDB, as reflected in the ERR bit and the DF bit in the ATA Status field.

Table 113 — ATA command results

CK_COND	Status field		Sense data returned
	ERR	DF	
0	0	0	No error, successful completion or command in progress. The SATL shall return GOOD status.
1			No error, successful completion or command in progress. The SATL shall terminate the command with CHECK CONDITION status with the sense key set to RECOVERED ERROR with the additional sense code set to ATA PASS-THROUGH INFORMATION AVAILABLE (see SPC-4). Descriptor format sense data shall include the ATA Status Return Descriptor (see 12.2.6)[a].
n/a	n/a	1	The ATA command completed with an error. The SATL shall terminate the command with CHECK CONDITION status with the sense key and additional sense code set as described in clause 11. Descriptor format sense data shall include the ATA Status Return Descriptor (see 12.2.6).
	1	0	

[a] This capability allows the host to retrieve the ATA register or field information with successful command completion by returning data in the ATA registers or fields.

ATA commands may return information in the ATA registers or the Shadow Command Block. The current ATA register information may be retrieved by requesting the ATA Status Return Descriptor issuing the ATA PASS-THROUGH (12) command or ATA PASS-THROUGH (16) command with the PROTOCOL field set to 15 (i.e., Return Response Information).

12.2.6 ATA Status Return descriptor

Table 114 shows the format of the ATA Return descriptor.

Each time the ATA Return descriptor is requested, the SATL shall read the ATA registers and return those values in the sense data as shown in table 114. If the sense data is for an ATA PASS-THROUGH (12) command or for the ATA PASS-THROUGH (16) command with the EXTEND bit set to zero, then the SATL shall return the 28-bit extended status and shall set the EXTEND bit to zero.

If the sense data is for an ATA PASS-THROUGH (16) command with the EXTEND bit set to one, then the SATL shall return the 48-bit extended status and shall set the EXTEND bit to one.

Table 114 — ATA Return descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	DESCRIPTOR CODE (09h)							
1	ADDITIONAL DESCRIPTOR LENGTH (0Ch)							
2	Reserved							EXTEND
3	ERROR							
4	SECTOR_COUNT (15:8)							
5	SECTOR_COUNT (7:0)							
6	LBA_LOW (15:8)							
7	LBA_LOW (7:0)							
8	LBA_MID (15:8)							
9	LBA_MID (7:0)							
10	LBA_HIGH (15:8)							
11	LBA_HIGH (7:0)							
12	DEVICE							
13	STATUS							

An EXTEND bit set to one indicates that the sense data is for an ATA PASS-THROUGH (16) command with the EXTEND bit set to one. An EXTEND bit set to zero indicate that the sense data is for an ATA PASS-THROUGH (16) command with the EXTEND bit set to zero or for an ATA PASS-THROUGH (12) command.

If the EXTEND bit is set to one, the SECTOR_COUNT (15:8) field, the LBA_LOW (15:8) field, the LBA_MID (15:8) field, and the LBA_HIGH (15:8) field are not returned in the fixed format sense data.

If the EXTEND bit is set to one, then the SECTOR_COUNT (7:0) field and SECTOR_COUNT (15:8) field indicate the ATA Sector Count. If the EXTEND bit is set to zero, then the SECTOR_COUNT (7:0) field indicates the ATA Sector Count and SECTOR_COUNT (15:8) field should be ignored.

If the EXTEND bit is set to one, then the LBA_LOW (7:0) field, LBA_MID (7:0) field, LBA_HIGH (7:0) field, LBA_LOW (15:8) field, LBA_MID (15:8) field, and LBA_HIGH (15:8) field specify the ATA LBA. If the EXTEND bit is set to zero, then the LBA_LOW (7:0) field, LBA_MID (7:0) field, and LBA_HIGH (7:0) field indicate the ATA LBA, and the LBA_LOW (15:8) field, LBA_MID (15:8) field, and LBA_HIGH (15:8) field shall be set to zero.

12.2.7 Fixed format sense data

Table 115 shows the fields returned in the fixed format sense data (see SPC-4) for the ATA PASS-THROUGH commands.

Table 115 — Fixed format sense data fields for the ATA PASS-THROUGH commands

Field	Descriptor or reference
VALID	Unspecified (see 3.4.2)
RESPONSE CODE	Unspecified (see 3.4.2)
FILEMARK	Set to zero
EOM	Set to zero
ILI	Set to zero
SENSE KEY	Unspecified (see 3.4.2)
INFORMATION	Table 116
ADDITIONAL SENSE LENGTH	Unspecified (see 3.4.2)
COMMAND-SPECIFIC INFORMATION	Table 117
ADDITIONAL SENSE CODE	Unspecified (see 3.4.2)
ADDITIONAL SENSE CODE QUALIFIER	Unspecified (see 3.4.2)
FIELD REPLACEABLE UNIT CODE	Unspecified (see 3.4.2)
SKSV	Unspecified (see 3.4.2)
SENSE-KEY SPECIFIC	Unspecified (see 3.4.2)
Additional sense bytes	Unspecified (see 3.4.2)

[a] SATLs compliant with previous versions of this standard return descriptor format sense data for the ATA PASS-THROUGH commands regardless of the setting of the D_SENSE bit.

Table 116 defines the INFORMATION field.

Table 116 — Fixed format sense data INFORMATION field for the ATA PASS-THROUGH commands

Byte\Bit	7	6	5	4	3	2	1	0
0	ERROR							
1	STATUS							
2	DEVICE							
3	SECTOR_COUNT (7:0)							

Table 117 defines the COMMAND-SPECIFIC INFORMATION field.

Table 117 — Fixed format sense data COMMAND-SPECIFIC INFORMATION field for the ATA PASS-THROUGH commands

Byte\Bit	7	6	5	4	3	2	1	0
0	EXTEND	SECTOR_COUNT UPPER NONZERO	LBA UPPER NONZERO	Reserved	LOG INDEX			
1	LBA_HIGH (7:0)							
2	LBA_MID (7:0)							
3	LBA_LOW (7:0)							

An EXTEND bit set to one indicates that the sense data is for an ATA PASS-THROUGH (16) command with the EXTEND bit set to one. An EXTEND bit set to zero indicates that the sense data is for an ATA PASS-THROUGH (16) command with the EXTEND bit set to zero or for an ATA PASS-THROUGH (12) command.

If the EXTEND bit is set to one, then the SECTOR_COUNT (15:8) field, the LBA_LOW (15:8) field, the LBA_MID (15:8) field; and the LBA_HIGH (15:8) field are not able to be returned in fixed format sense data.

If the LBA UPPER NONZERO bit is set to one, then one or more of the LBA_LOW (15:8) field, the LBA_MID (15:8) field, and the LBA_HIGH (15:8) field returned by the ATA device were not set to 00h. If the LBA UPPER NONZERO bit is set to zero, then the LBA_LOW (15:8) field, the LBA_MID (15:8) field, and the LBA_HIGH (15:8) field returned by the ATA device were each set to 00h.

If the SECTOR_COUNT UPPER NONZERO bit is set to one, then the SECTOR_COUNT (15:8) field returned by the ATA device was not set to 00h. If the SECTOR_COUNT UPPER NONZERO bit is set to zero, then the SECTOR_COUNT (15:8) field returned by the ATA device was set to 00h.

A LOG INDEX field set to a nonzero value indicates that the device server has logged the descriptor format sense data for the command for retrieval via the ATA PASS-THROUGH Results log page (see 12.x.2) with a parameter code set to log index minus 1 (e.g. log index 1h corresponds to parameter code 0h, and log index Fh corresponds to parameter code Eh). A LOG INDEX field set to 0h indicates that the device server has not logged the descriptor format sense data for the command for retrieval via the ATA PASS-THROUGH Results log page.

The device server:

a) should log the descriptor format sense data if the LBA UPPER NONZERO bit is set to one or the SECTOR_COUNT UPPER NONZERO bit is set to one;
b) shall not log the descriptor format sense data if the LBA UPPER NONZERO bit is set to zero and the SECTOR_COUNT UPPER NONZERO bit is set to zero;
c) shall select the log index as the previously reported log index plus one, wrapping from Fh to 0h.

12.3 SAT-specific mode pages

12.3.1 SAT-specific mode pages overview

This subclause describes mode pages that the SATL may implement that are unique to the SCSI / ATA Translation standard. These mode pages are for use by the SATL, are shown in table 118 and are described in this subclause.

Table 118 — SCSI / ATA Translation specific mode pages

PAGE CODE	SUBPAGE CODE	Mode page name
0Ah	F1h	PATA Control[a]
1Ah	F1h	ATA Power Condition

[a] Support of the PATA Control mode page is mandatory for SATLs implementing a PATA interface.

12.3.2 PATA Control mode page

The PATA Control mode page provides PATA specific controls for a SATL to configure the underlying PATA host and to understand what parameters are communicated to the PATA device to ensure proper communication for specific transfer rates. This standard specifies the mode parameters that are provided for this mode page.

SATL implementations that support the attachment of PATA devices shall support this mode page. The SATL should allow application clients to configure alternate PATA timings using the MODE SELECT command.

Table 119 shows the PATA Control mode page.

Table 119 — PATA Control mode page

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (0Ah)					
1	SUBPAGE CODE (F1h)							
2	(MSB)	PAGE LENGTH (0004h)						
3								(LSB)
4	Reserved	MWDMA[a] bits			Reserved		PIO[b] bits	
		MWD2	MWD1	MWD0			PIO4	PIO3
5	Reserved	UDMA[c] bits						
		UDMA6	UDMA5	UDMA4	UDMA3	UDMA2	UDMA1	UDMA0
6	Reserved							
7								

[a] The Multi-Word Direct Memory Access (MWDMA) bits specify a number of hardware-assisted data transfer modes defined in ATA8-APT.
[b] PIO stands for Programmed Input and Output and the PIOx bits specify transfer modes performed under program control defined in ATA8-APT.
[c] The Ultra Direct Memory Access (UDMA) bits represent a number of hardware-assisted data transfer modes defined in ATA8-APT.

SATL implementations may save the state of the timing parameters defined in this mode page.

Application clients may use the MODE SENSE command for changeable values to determine the underlying ATA host support for a given ATA timing mode. The SATL shall support changeable mode parameters for this mode page.

When processing a MODE SENSE command, the SATL shall set the PIO3 bit and PIO4 bit as shown in table 120 to identify the configured PIO mode.

Table 120 — PIO modes

PIO4	PIO3	PIO mode
0	0	Reserved
0	1	The ATA host shall use PIO mode 3 transfers.
1	0	The ATA host shall use PIO mode 4 transfers.
1	1	Reserved

When changeable values are requested, the PIO3 bit and the PIO4 bit indicate if the underlying ATA host supports those transfer modes. The PIO3 bit shall be set to one if the ATA host supports PIO mode 3. The PIO3 bit and the PIO4 bit shall be set to one if the ATA host supports PIO mode 4.

If the SATL receives a MODE SELECT command and the PIO bits specify a change from the current setting, then the SATL shall configure the ATA host to use the new PIO transfer rate, if supported. If the application client requests a PIO setting that the ATA device does not support, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The MWD0 bit, the MWD1 bit, and the MWD2 bit are collectively referred to as the MWDMA bits. If the ATA host in the SATL is currently configured to use multiword DMA (MWDMA), then the MWDMA bits are used to determine what mode is currently being used, what modes are supported by the ATA host, and control of the MWDMA mode.

If the SATL receives a MODE SENSE command requesting the current values of the PATA Control mode page, then the MWD0 bit shall be set to one by the SATL when the host and device are configured to use MWDMA mode 0. The MWD1 bit shall be set to one by the SATL when the host and device are configured to use MWDMA mode1. The MWD2 bit shall be set to one by the SATL when the host and device are configured to use MWDMA mode 2.

If the SATL receives a MODE SENSE command requesting the changeable values of the PATA Control mode page, then the MWD0 bit shall be set to one if the ATA host supports MWDMA mode 0. The MWD1 bit and MWD0 bit shall each be set to one if the ATA host supports MWDMA mode 1. The MWD2 bit, the MWD1 bit, and the MWD0 bit shall be each be set to one if the ATA host supports MWDMA mode 2.

Table 121 specifies values set by the SATL in the MWD0 bit, the MWD1 bit, and the MWD2 bit for current and changeable MWDMA settings.

Table 121 — MWDMA modes reported by MODE SENSE

MWDMA[a] bits			ATA host and device shared configuration settings returned as current values	ATA host support returned as changeable values
MWD2	MWD1	MWD0		
0	0	0	Configured not to use multiword DMA	Illegal combination
1	0	0	Configured to use MWDMA mode 1	
0	1	0	Configured to use MWDMA mode 2	
1	1	0	Configured to use MWDMA modes 1 and 2	
0	0	1	Configured to use MWDMA mode 0	MWDMA mode 0 supported
1	0	1	Configured to use MWDMA modes 0 and 2	Illegal combination
0	1	1	Configured to use MWDMA modes 0 and 1	MWDMA mode 1 supported
1	1	1	Configured to use MWDMA modes 0, 1 and 2	MWDMA mode 2 supported

[a] If the application client attempts to set a MWDMA mode that is not supported by the ATA host environment, then the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

If the SATL receives a MODE SELECT command and the MWDMA bits specify a change from the current settings, then the SATL shall send an ATA SET FEATURES - Set transfer mode (i.e., Features register set to 03h) command to the ATA device to set the MWDMA mode on the ATA device to the requested state, and then:

a) if the ATA SET FEATURES command completes with an error, then the SATL shall:
   1) not change any host transfer modes; and
   2) complete the MODE SELECT command with a CHECK CONDITION status with the sense key set to ABORTED COMMAND with the additional sense code set to ATA DEVICE FAILED SET FEATURES;

   or

b) if the ATA SET FEATURES command completes without error, then the SATL shall:
   1) configure the ATA host to communicate with the device at the requested MWDMA transfer rate; and
   2) complete the MODE SELECT command with GOOD status.

The MWDMA bits values used to configure ATA hosts and ATA devices using the MODE SELECT command have the same meaning as the MWDMA bits values returned by the MODE SENSE command when current values are requested as shown in table 121.

If the SATL receives a request to set a MWDMA mode that is not supported by the ATA host or the attached PATA device, then the SATL shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The UDMA0 bit, the UDMA1 bit, the UDMA2 bit, the UDMA3 bit, the UDMA4 bit, the UDMA5 bit, and the UDMA6 bit are collectively referred to as the UDMA bits, and are used to determine support for, current use of, and control of Ultra DMA (UDMA) transfer rates on the ATA host and device. The SATL shall determine the highest UDMA mode supported as being the lower of the ATA host maximum transfer mode and the device maximum transfer mode.

NOTE 11 - The ATA device returns the UDMA transfer mode specified in ATA IDENTIFY DEVICE data, word 88 bits 6:0 (see ATA8-ACS).

If the SATL receives a MODE SENSE command requesting the changeable values of the PATA Control mode page, then the UDMA bits shall be set according to table 122.

Table 122 — UDMA bits requirements for changeable MODE SENSE parameters

UDMA6	UDMA5	UDMA4	UDMA3	UDMA2	UDMA1	UDMA0	Highest UDMA mode supported
0	0	0	0	0	0	0	UDMA Unsupported
0	0	0	0	0	0	1	0
0	0	0	0	0	1	1	1
0	0	0	0	1	1	1	2
0	0	0	1	1	1	1	3
0	0	1	1	1	1	1	4
0	1	1	1	1	1	1	5
1	1	1	1	1	1	1	6

If the SATL receives a MODE SENSE command requesting the current values of the PATA Control mode page, then the SATL shall set the UDMA bits as defined in table 123. Only one of the UDMA bits shall be set to one at any time for such a request. If UDMA is not the current DMA transfer mode, then all the UDMA bits shall be set to zero. If a UDMA transfer mode is being used, then all of the MWDMA bits shall be set to zero.

Table 123 — UDMA for current MODE SENSE settings

UDMA bit	Value	Description
UDMA0	0	ATA host and device are not communicating using UDMA Mode 0
	1	ATA host and device are communicating using UDMA Mode 0
UDMA1	0	ATA host and device are not communicating using UDMA Mode 1
	1	ATA host and device are communicating using UDMA Mode 1
UDMA2	0	ATA host and device are not communicating using UDMA Mode 2
	1	ATA host and device are communicating using UDMA Mode 2
UDMA3	0	ATA host and device are not communicating using UDMA Mode 3
	1	ATA host and device are communicating using UDMA Mode 3
UDMA4	0	ATA host and device are not communicating using UDMA Mode 4
	1	ATA host and device are communicating using UDMA Mode 4
UDMA5	0	ATA host and device are not communicating using UDMA Mode 5
	1	ATA host and device are communicating using UDMA Mode 5
UDMA6	0	ATA host and device are not communicating using UDMA Mode 6
	1	ATA host and device are communicating using UDMA Mode 6

When the SATL receives a MODE SELECT command and the UDMA bits request a change in the UDMA transfer rate, then the SATL shall:

1) if the SET FEATURES command completes with an error, then the SATL shall:
   A) not change any host transfer modes; and
   B) complete the MODE SELECT command with a CHECK CONDITION status with the sense key set to ABORTED COMMAND with the additional sense code set to ATA DEVICE FAILED SET FEATURES;

   or

2) if the SET FEATURES command completes without error, then the SATL shall:
   A) configure the ATA host to communicate with the device at the requested UDMA transfer rate; and
   B) complete the MODE SELECT command with GOOD status.

If the application client attempts to set a mode that the ATA host or the ATA device does not support, then the SATL shall terminate the MODE SELECT command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

12.3.3 ATA Power Condition mode page

The ATA Power Condition mode page provides ATA specific controls for a SATL to configure ATA specific power management functions.

Table 124 shows the ATA Power Condition mode page.

Table 124 — ATA Power Condition mode page

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (1Ah)					
1	SUBPAGE CODE (F1h)							
2	(MSB)	PAGE LENGTH (000Ch)						
3								(LSB)
4	Reserved							
5	Reserved							APMP
6	APM VALUE							
7	Reserved							
15								

When processing a MODE SELECT command, if the APMP bit is set to zero, then the SATL shall ignore the APM VALUE field.

When processing a MODE SELECT command, if the APMP bit is set to one, then the SATL shall alter the ATA APM mode by issuing an ATA SET FEATURES command. If the APM VALUE field contains a non-zero value, then the ATA SET FEATURES – Enable/disable the APM feature set (i.e., subcommand 05h) command shall be sent and the APM VALUE field shall be used to set the power management level (i.e., COUNT field). If the APM VALUE field contains a zero, then the ATA SET FEATURES – Disable the APM feature set (i.e., subcommand 85h) command shall be sent.

If the ATA SET FEATURES command completes with an error, then the SATL shall terminate the MODE SELECT command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

When processing a MODE SENSE command, the SATL shall determine if ATA APM mode is enabled by verifying that ATA IDENTIFY DEVICE data word 83, bit 3 is set to one, and that ATA IDENTIFY DEVICE data word 86, bit 3 is also set to one. If ATA APM mode is not enabled, then the APMP bit shall be set to zero. If ATA APM mode is enabled, then the APMP bit shall be set to one and the APM VALUE field shall contain the value from ATA IDENTIFY DEVICE word 91 bits (7:0).

12.4 SAT-specific VPD pages

12.4.1 SAT-specific VPD pages overview

This subclause defines VPD pages specific to SAT implementations.

12.4.2 ATA Information VPD page

12.4.2.1 ATA Information VPD page overview

The ATA Information VPD page contains:

a) information about the SATL;
b) Signature of the ATA or ATAPI device; and
c) ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data from the ATA or ATAPI device.

Some SATLs may modify ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data. If a SCSI application client requires the unmodified ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data, then the ATA PASS-THROUGH command (see 12.2) should be used to retrieve the unmodified ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data.

Table 125 defines the ATA Information VPD page.

Table 125 — ATA Information VPD page

Byte\Bit	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (89h)							
2	(MSB)			PAGE LENGTH (238h)				
3								(LSB)
4	Reserved							
7								
8	SAT VENDOR IDENTIFICATION							
15								
16	SAT PRODUCT IDENTIFICATION							
31								
32	SAT PRODUCT REVISION LEVEL							
35								
36	Device signature (see 12.4.2.2)							
55								
56	COMMAND CODE							
57	Reserved							
59								
60	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data (see 12.4.2.3)							
571								

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field shall be set as described in 8.1.2.

The SAT VENDOR IDENTIFICATION field shall contain an 8-byte ASCII string identifying the vendor of the SATL. The data shall be left aligned within the field. The vendor identification string shall be one assigned by INCITS for use in the Standard INQUIRY data VENDOR IDENTIFICATION field. A list of assigned vendor identification strings is in SPC-4 and on the T10 web site (http://www.t10.org).

The SAT PRODUCT IDENTIFICATION field shall contain sixteen bytes of ASCII data as defined by the vendor of the SATL. The data shall be left-aligned within the field.

The SAT PRODUCT REVISION LEVEL field shall contain four bytes of ASCII data as defined by the vendor of the SATL. The data shall be left-aligned within the field.

The ATA device signature is described in 12.4.2.2.

The COMMAND CODE field contains the of the ATA command code used to retrieve the data in the ATA IDENTIFY DEVICE or ATA IDENTIFY PACKET DEVICE DATA field. The possible command codes are:

a) ECh for an ATA IDENTIFY DEVICE command (i.e., for an ATA device);
b) A1h for an ATA IDENTIFY PACKET DEVICE command (i.e., for an ATAPI device); or
c) 00h for other device types.

The ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data is described in 12.4.2.3.

12.4.2.2 ATA device signature

The ATA device signature shall contain the contents of the task file registers after the last power-on reset, hardware reset, software reset, or ATA EXECUTE DEVICE DIAGNOSTIC command. The ATA device signature shall follow the format of the initial SATA Device-to-Host Register FIS (see SATA-2.6). Table 126 shows the ATA device signature.

Table 126 — ATA device signature

Byte\Bit	7	6	5	4	3	2	1	0
0	TRANSPORT IDENTIFIER							
1	Reserved	INTERRUPT/ Reserved[a]	Reserved		PM PORT / Reserved[a]			
2	STATUS[b]							
3	ERROR[b]							
4	LBA (7:0)[b]							
5	LBA (15:8)[b]							
6	LBA (23:16)[b]							
7	DEVICE[b]							
8	LBA (31:24)[b]							
9	LBA (39:32)[b]							
10	LBA (47:40)[b]							
11	Reserved							
12	SECTOR COUNT (7:0)[b]							
13	SECTOR COUNT (15:8)[b]							
14	Reserved							
19								

[a] The INTERRUPT bit and the PM PORT field are defined only if the TRANSPORT IDENTIFIER field is set to 34h (see SATA-2.6). Otherwise the INTERRUPT field and the PM PORT field are reserved.
[b] These fields are fields with the same names defined in ATA8-ACS.

The TRANSPORT IDENTIFIER field is defined in table 127.

Table 127 — TRANSPORT IDENTIFIER field

Code	Transport
00h	PATA (see ATA8-APT)
34h	SATA (see SATA-2.6)
All others	Reserved

The INTERRUPT bit corresponds to the “I” bit (i.e., bit 14 of dword 0) of the Register Device-to-Host FIS (see SATA-2.6).

All the remaining fields within the ATA device signature are defined in ATA8-APT and SATA-2.6.

12.4.2.3 ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data

If the command is an ATA IDENTIFY DEVICE command and the command completes without error, then the ATA IDENTIFY DEVICE or ATA IDENTIFY PACKET DEVICE DATA field shall contain the ATA IDENTIFY DEVICE data (ATA8-ACS).

If the command is an ATA IDENTIFY PACKET DEVICE command, and the command completes without error, then the ATA IDENTIFY DEVICE or ATA IDENTIFY PACKET DEVICE DATA field shall contain the IDENTIFY PACKET DEVICE data (see ATA8-ACS).

The ATA IDENTIFY DEVICE or ATA IDENTIFY PACKET DEVICE DATA field shall contains 512 bytes of 00h if:

a) the command is an ATA IDENTIFY DEVICE command or an ATA IDENTIFY PACKET DEVICE command and the command completes with an error; or
b) the command code is 00h (i.e., some other device type).

The data shall be presented with byte preservation (i.e., ATA byte n maps to SCSI byte n), as shown in table 128.

Table 128 — ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data

Byte	Contents
0	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 0 bits 7:0 (i.e., byte 0)
1	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 0 bits 15:8 (i.e., byte 1)
2	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 1 bits 7:0 (i.e., byte 2)
3	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 1 bits 15:8 (i.e., byte 3)
...	...
510	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 255 bits 7:0 (i.e., the signature byte of the Integrity word, see ATA8-ACS)
511	ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data word 255 bits 15:8 (i.e., the checksum byte of the Integrity word, see ATA8-ACS)

NOTE 12 - Although the Serial number field (i.e., words 19:10), Firmware revision field (i.e., words 26:23), and Model number field (i.e., words 46:27) contain ASCII characters, every other byte is swapped within them (see ATA8-ACS) (e.g., the Serial number field is interpreted as: {word 10 bits 15:8, word 10 bits 7:0, word 11 bits 15:8, word 11 bits 7:0,...}, which corresponds to these bytes in the IDENTIFY DEVICE OR IDENTIFY PACKET DEVICE DATA field: {byte 21, byte 20, byte 23, byte 22, etc.}).

Since some of the fields within the ATA IDENTIFY DEVICE data or ATA IDENTIFY PACKET DEVICE data may change depending on the state of the ATA device, the SATL shall resend the ATA IDENTIFY DEVICE command or ATA IDENTIFY PACKET DEVICE command to retrieve updated data whenever the ATA Information VPD page is requested.

12.5 SAT-specific security protocol parameters

12.5.1 ATA Device Server Password security protocol

12.5.1.1 SECURITY PROTOCOL IN command

12.5.1.1.1 SECURITY PROTOCOL IN command overview

The SECURITY PROTOCOL IN command is used by the application client to cause the SATL to return ATA Security feature set data extracted from the ATA IDENTIFY DEVICE data from the ATA device. See ATA8-ACS for a description of the ATA Security feature set.

If the SECURITY PROTOCOL field is set to EFh in a SECURITY PROTOCOL IN command, then the SECURITY PROTOCOL SPECIFIC field shall be set to zero. All other values of the SECURITY PROTOCOL SPECIFIC field are reserved.

The INC_512 bit shall be set to zero. If a SECURITY PROTOCOL IN command is received with the INC_512 bit is set to one, then the SECURITY PROTOCOL IN command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

All other CDB fields for SECURITY PROTOCOL IN command shall meet the requirements stated in SPC-4.

12.5.1.1.2 SECURITY PROTOCOL IN parameter data

Table 129 defines the parameter data sent in response to a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to EFh.

Table 129 — SECURITY PROTOCOL IN parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	PARAMETER LIST LENGTH (0Dh)							
2	(MSB)	SECURITY ERASE TIME						
3								(LSB)
4	(MSB)	ENHANCED SECURITY ERASE TIME						
5								(LSB)
6	(MSB)	MASTER PASSWORD IDENTIFIER						
7								(LSB)
8	Reserved							MAXSET
9	Reserved		EN_ER_SUP	PWCNTEX	FROZEN	LOCKED	S_ENABLD	S_SUPRT
10	Reserved							
15								

The value in the SECURITY ERASE TIME field indicates the time required by the ATA device to complete its security erase procedure in normal mode. The SATL shall set the least significant byte of the SECURITY ERASE TIME field to the ATA IDENTIFY DEVICE data word 89 bits (0:7) and the most significant byte of the SECURITY ERASE TIME field to 00h.

The value in the ENHANCED SECURITY ERASE TIME field indicates the time required by the ATA device to complete its security erase procedure in enhanced mode. The SATL shall set the least significant byte of the ENHANCED SECURITY ERASE TIME field to the ATA IDENTIFY DEVICE data word 90 bits (0:7) and the most significant byte of the ENHANCED SECURITY ERASE TIME field to 00h.

The SATL shall set the MASTER PASSWORD IDENTIFIER field to the ATA IDENTIFY DEVICE data word 82.

If the ATA IDENTIFY DEVICE data word 128 bit 8 is set to zero, then the SATL shall set the master password capability setting (MAXSET) bit to zero. If the ATA IDENTIFY DEVICE data word 128 bit 8 is set to one, then the SATL shall set the master password capability setting (MAXSET) bit to one.

If the ATA IDENTIFY DEVICE data word 128 bit 5 is set to zero, then the SATL shall set the enhanced erase mode supported (EN_ER_SUP) bit to zero. If the ATA IDENTIFY DEVICE data word 128 bit 5 is set to one, then the SATL shall set the enhanced erase mode supported (EN_ER_SUP) bit to one.

If the ATA IDENTIFY DEVICE data word 128 bit 4 is set to zero, then the SATL shall set the password attempt counter exceeded (PWCNTEX) bit to zero. If the ATA IDENTIFY DEVICE data word 128 bit 4 is set to one, then the SATL shall set the password attempt counter exceeded (PWCNTEX) bit to one.

If the ATA IDENTIFY DEVICE data word 128 bit 3 is set to zero, then the SATL shall set the frozen state (FROZEN) bit to zero. If the ATA IDENTIFY DEVICE data word 128 bit 3 is set to one, then the SATL shall set the frozen state (FROZEN) bit to one.

If the ATA IDENTIFY DEVICE data word 128 bit 2 is set to zero, then the SATL shall set the locked state (LOCKED) bit to zero. If the ATA IDENTIFY DEVICE data word 128 bit 2 is set to one, then the SATL shall set the locked state (LOCKED) bit to one.

If the ATA IDENTIFY DEVICE data word 85 bit 1 is set to zero, then the SATL shall set the ATA Security feature set enabled (S_ENABLD) bit to zero. If the ATA IDENTIFY DEVICE data word 85 bit 1 is set to one, then the SATL shall set the ATA Security feature set enabled (S_ENABLD) bit to one. Enabling of this bit is based on setting of the user password via a set password function (see 12.5.1.2.1).

If the ATA IDENTIFY DEVICE data word 82 bit 1 is set to zero, then the SATL shall set the ATA Security feature set supported (S_SUPRT) bit to zero. If the ATA IDENTIFY DEVICE data word 82 bit 1 is set to one, then the SATL shall set the ATA Security feature set supported (S_SUPRT) bit to one.

12.5.1.1.3 SCSI commands allowed in the presence of various security modes

Certain commands may be allowed or conflict depending on the security mode setting that is in effect for an ATA device.

There are three possible modes:

a) security locked;
b) security unlocked or security disabled; and
c) security frozen.

If a SATL receives a command that is allowed for the current security mode setting of the ATA device, then the SATL translates the command as defined in this standard and sends it to the ATA device. If a SATL receives a command that conflicts with the current security mode setting of the ATA device, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to ATA SECURITY CONFLICT.

Table 130 shows the commands defined in SPC-4 and whether each command is allowed or conflicts depending on the security setting that is in effect for an ATA device. If a command in table 130 is not implemented by the SATL, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE.

Table 130 — SPC commands allowed in the presence of various ATA security modes (part 1 of 3)

Command	Locked	Unlocked or Disabled	Frozen
ACCESS CONTROL IN[a]	Allowed	Allowed	Allowed
ACCESS CONTROL OUT[a]	Allowed	Allowed	Allowed
CHANGE ALIASES[a]	Allowed	Allowed	Allowed
EXTENDED COPY[a]	Conflict	Allowed	Allowed
INQUIRY	Allowed	Allowed	Allowed
LOG SELECT	Allowed[b]	Allowed	Allowed

[a] ATA SECURITY CONFLICT shall not be returned for this command.
[b] Allowed unless otherwise specified.

Table 130 — SPC commands allowed in the presence of various ATA security modes (part 2 of 3)

Command	Locked	Unlocked or Disabled	Frozen
LOG SENSE[a]	Allowed	Allowed	Allowed
MANAGEMENT PROTOCOL IN[a]	Allowed	Allowed	Allowed
MANAGEMENT PROTOCOL OUT[a]	Allowed	Allowed	Allowed
MODE SELECT(6) / MODE SELECT(10)			
All mode pages	Allowed	Allowed	Allowed
MODE SENSE(6) / MODE SENSE(10)	Allowed	Allowed	Allowed
PERSISTENT RESERVE IN[a]	Allowed	Allowed	Allowed
PERSISTENT RESERVE OUT			
REGISTER[a]	Allowed	Allowed	Allowed
RESERVE[a]	Allowed	Allowed	Allowed
RELEASE[a]	Allowed	Allowed	Allowed
CLEAR[a]	Allowed	Allowed	Allowed
PREEMPT[a]	Allowed	Allowed	Allowed
PREEMPT AND ABORT[a]	Allowed	Allowed	Allowed
REGISTER AND IGNORE EXISTING KEY[a]	Allowed	Allowed	Allowed
REGISTER AND MOVE[a]	Allowed	Allowed	Allowed
READ ATTRIBUTE[a]	Allowed	Allowed	Allowed
READ BUFFER	Allowed	Allowed	Allowed
READ MEDIA SERIAL NUMBER[a]	Allowed	Allowed	Allowed
RECEIVE COPY RESULTS[a]	Allowed	Allowed	Allowed
RECEIVE DIAGNOSTIC RESULTS[a]	Allowed	Allowed	Allowed
RELEASE(6) / RELEASE(10)[a]	Allowed	Allowed	Allowed
REPORT ALIASES[a]	Allowed	Allowed	Allowed
REPORT IDENTIFYING INFORMATION[a]	Allowed	Allowed	Allowed
REPORT LUNS[a]	Allowed	Allowed	Allowed
REPORT PRIORITY[a]	Allowed	Allowed	Allowed
REPORT SUPPORTED OPERATION CODES[a]	Allowed	Allowed	Allowed
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS[a]	Allowed	Allowed	Allowed
REPORT TARGET PORT GROUPS[a]	Allowed	Allowed	Allowed

[a] ATA SECURITY CONFLICT shall not be returned for this command.
[b] Allowed unless otherwise specified.

Table 130 — SPC commands allowed in the presence of various ATA security modes (part 3 of 3)

Command	Locked	Unlocked or Disabled	Frozen
REPORT TIMESTAMP[a]	Allowed	Allowed	Allowed
REQUEST SENSE	Allowed	Allowed	Allowed
RESERVE(6) / RESERVE(10)[a]	Allowed	Allowed	Allowed
SECURITY PROTOCOL IN[a]	Allowed	Allowed	Allowed
SECURITY PROTOCOL OUT			
Tape Data Encryption[a]	Conflict	Conflict	Conflict
Authentication in Host Attachments of Transient Storage Devices[a]	Conflict	Conflict	Conflict
Device Server Password Security	Allowed	Allowed	Conflict
IEEE 1667[a]	Conflict	Conflict	Conflict
TCG[a]	Conflict	Conflict	Conflict
SEND DIAGNOSTIC	Allowed	Allowed	Allowed
SET IDENTIFYING INFORMATION[a]	Allowed	Allowed	Allowed
SET PRIORITY[a]	Allowed	Allowed	Allowed
SET TARGET PORT GROUPS[a]	Allowed	Allowed	Allowed
SET TIMESTAMP[a]	Allowed	Allowed	Allowed
TEST UNIT READY	Allowed	Allowed	Allowed
WRITE ATTRIBUTE[a]	Allowed	Allowed	Allowed
WRITE BUFFER	Allowed	Allowed	Allowed

[a] ATA SECURITY CONFLICT shall not be returned for this command.
[b] Allowed unless otherwise specified.

Table 131 shows the commands defined in SBC-3 and whether each command is allowed or conflicts depending on the security setting that is in effect for an ATA device. If a command in table 131 is not implemented by the SATL, then the SATL shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID COMMAND OPERATION CODE.

12.5.1.2 SECURITY PROTOCOL OUT command

12.5.1.2.1 SECURITY PROTOCOL OUT command overview

The SECURITY PROTOCOL OUT command is used by an application client to send ATA Security feature set commands and data to the ATA device. See ATA8-ACS for a description of the ATA Security feature set and all of the functions defined in this standard.

Table 131 — SBC commands allowed in the presence of various ATA security modes

Command	Locked	Unlocked or Disabled	Frozen
FORMAT UNIT	Conflict	Allowed	Allowed
ORWRITE[a]	Conflict	Allowed	Allowed
PRE-FETCH (10) / (16)[a]	Conflict	Allowed	Allowed
PREVENT ALLOW MEDIUM REMOVAL (Prevent=0)[a]	Conflict	Allowed	Allowed
PREVENT ALLOW MEDIUM REMOVAL (Prevent<>0)[a]	Conflict	Allowed	Allowed
READ (6) / (10) / (12) / (16) / (32)	Conflict	Allowed	Allowed
READ CAPACITY (10) / (16)	Allowed	Allowed	Allowed
READ DEFECT DATA (10) / (12)[a]	Conflict	Allowed	Allowed
READ LONG (10) / (16)[a]	Conflict	Allowed	Allowed
REASSIGN BLOCKS	Conflict	Allowed	Allowed
START STOP UNIT	Allowed	Allowed	Allowed
SYNCHRONIZE CACHE (10) / (16)	Conflict	Allowed	Allowed
VERIFY (10) / (12) / (16) / (32)	Conflict	Allowed	Allowed
WRITE (6) / (10) / (12) / (16) / (32)	Conflict	Allowed	Allowed
WRITE AND VERIFY (10) / (12) / (16) / (32)	Conflict	Allowed	Allowed
WRITE LONG (10) / (16)[a]	Conflict	Allowed	Allowed
WRITE SAME (10) / (16) / (32)	Conflict	Allowed	Allowed
XDREAD (10) / (32)	Conflict	Allowed	Allowed
XDWRITE (10) / (32)	Conflict	Allowed	Allowed
XDWRITEREAD (10) / (32)	Conflict	Allowed	Allowed
XPWRITE (10) / (32)	Conflict	Allowed	Allowed

[a] ATA SECURITY CONFLICT shall not be returned for this command.

When the SECURITY PROTOCOL field is set to EFh in a SECURITY PROTOCOL OUT command, the SECURITY PROTOCOL SPECIFIC field specifies the ATA command that the SATL shall send to the ATA device(see table 132).

Table 132 — SECURITY PROTOCOL SPECIFIC field

SECURITY PROTOCOL SPECIFIC field	Description	ATA command processing reference	Parameter data reference
0000h	Reserved		
0001h	Set password	ATA SECURITY SET PASSWORD	12.5.1.2.2
0002h	Unlock	ATA SECURITY UNLOCK	12.5.1.2.3
0003h	Erase prepare	ATA SECURITY ERASE PREPARE	No data is transferred
0004h	Erase unit	ATA SECURITY ERASE UNIT	12.5.1.2.4
0005h	Freeze lock	ATA SECURITY FREEZE LOCK	No data is transferred
0006h	Disable password	ATA SECURITY DISABLE PASSWORD	12.5.1.2.5
0007h through FFFFh	Reserved		

The INC_512 bit shall be set to zero. If a SECURITY PROTOCOL OUT command is received with the INC_512 bit is set to one, then the SECURITY PROTOCOL OUT command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

All other CDB fields for the SECURITY PROTOCOL OUT command shall meet the requirements stated in SPC-4.

12.5.1.2.2 Set password parameter list

If the SECURITY PROTOCOL SPECIFIC field is set to 0001h in the SECURITY PROTOCOL OUT CDB, then the TRANSFER LENGTH field in the CDB shall be set to 24h. Table 133 defines the parameter list for the SECURITY PROTOCOL OUT command when the SECURITY PROTOCOL SPECIFIC field is set to 0001h (set password).

Table 133 — Set password parameter list

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							MAXLVL
1	Reserved							MSTRPW
2	(MSB)	PASSWORD						
33								(LSB)
34	(MSB)	MASTER PASSWORD IDENTIFIER						
35								(LSB)

The SATL shall copy the contents of the MAXLVL bit to word 0, bit 8 of the ATA SECURITY SET PASSWORD data.

The SATL shall copy the contents of the MSTRPW bit to word 0, bit 0 of the ATA SECURITY SET PASSWORD data. If the mstrpw field is set to one, the SATL shall copy the contents of the MASTER PASSWORD IDENTIFIER field to the ATA SECURITY SET PASSWORD data.

The SATL shall copy the contents of the PASSWORD field to words (16:1) of the ATA SECURITY SET PASSWORD data.

12.5.1.2.3 Unlock parameter list

If the SECURITY PROTOCOL SPECIFIC field is set to 0002h in the SECURITY PROTOCOL OUT CDB, then the TRANSFER LENGTH field in the CDB shall be set to 24h. Table 134 defines the parameter list for the SECURITY PROTOCOL OUT command when the SECURITY PROTOCOL SPECIFIC field is set to 0002h (unlock).

Table 134 — Unlock parameter list

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							MSTRPW
2	(MSB)							
33				PASSWORD				(LSB)
34								
35				Reserved				

The SATL shall copy the MSTRPW bit to word 0, bit 0 of the ATA SECURITY UNLOCK data.

The SATL shall copy the PASSWORD field to words 16:1 of the ATA SECURITY UNLOCK data.

12.5.1.2.4 Erase unit parameter list

If the SECURITY PROTOCOL SPECIFIC field is set to 0004h in the SECURITY PROTOCOL OUT CDB, then the TRANSFER LENGTH field in the CDB shall be set to 24h. Table 135 defines the parameter list for the SECURITY PROTOCOL OUT command when the SECURITY PROTOCOL SPECIFIC field is set to 0004h (erase unit).

Table 135 — Erase unit parameter list

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							EN_ER
1	Reserved							MSTRPW
2	(MSB)							
33				PASSWORD				(LSB)
34								
35				Reserved				

The SATL shall copy the EN_ER bit to word 0, bit 1 of the ATA SECURITY ERASE UNIT data.

The SATL shall copy the MSTRPW bit to word 0, bit 0 of the ATA SECURITY ERASE UNIT data.

The SATL shall copy the password field to words 16:1 of the ATA SECURITY ERASE UNIT data.

12.5.1.2.5 Disable password parameter list

If the SECURITY PROTOCOL SPECIFIC field is set to 0006h in the SECURITY PROTOCOL OUT CDB, then the TRANSFER LENGTH field in the CDB shall be set to 24h. Table 136 defines the parameter list for the SECURITY PROTOCOL OUT command when the SECURITY PROTOCOL SPECIFIC field is set to 0006h (disable password).

Table 136 — Disable password parameter list

Byte \ Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved							MSTRPW
2	(MSB)	PASSWORD						
33								(LSB)
34		Reserved						
35								

The SATL shall copy the MSTRPW bit to word 0, bit 0 of the ATA SECURITY DISABLE PASSWORD data.

The SATL shall copy the PASSWORD field to words 16:1 of the ATA SECURITY DISABLE PASSWORD data.

12.6 SAT-specific log pages

12.6.1 SAT-specific log pages overview

This subclause describes log pages that the SATL may implement that are unique to the SCSI / ATA Translation standard. These log pages are for use by the SATL, are shown in table 137, and are described in this subclause.

Table 137 — SCSI / ATA Translation specific log pages

PAGE CODE	SUBPAGE CODE	Log page name	Reference
16h	00h	ATA PASS-THROUGH Results log page	12.6.2

12.6.2 ATA PASS-THROUGH Results log page

The ATA PASS-THROUGH Results log page reports descriptor format sense data for ATA PASS-THROUGH commands that were terminated with CHECK CONDITION status by a device server that returned fixed format sense data and was not able to return the complete set of information (see 12.2.6).

This log page uses the binary list parameter format defined in SPC-4.

The number of log parameters reported shall less than or equal to 15.

The PARAMETER CODE field of each log parameter indicates the value of the LOG INDEX field minus one returned in fixed format sense data (e.g. parameter code 0h corresponds to log index 1h, and parameter code Eh corresponds to log index Fh) (see 12.2.6). The device server shall support log parameter codes 0h through Eh.

The FORMAT AND LINKING field of each log parameter shall be set to 11b, indicating that the parameters are binary format list parameters. The values of the bits and fields in the parameter control byte for binary format list parameters are described in SPC-4.

The PARAMETER LENGTH field of each log parameter is defined in SPC-4.

The PARAMETER VALUE field of each log parameter indicates the descriptor format sense data.

Annex A
(normative)

SCSI to ATAPI command transmission

A.1 Introduction

This annex specifies the method of transmission of SCSI commands to an ATAPI device.

A.2 ATAPI device model

An ATAPI device operates by using the ATA PACKET command, in order to transmit a SCSI CDB to the device. In addition to the SCSI command set supported by the device, the ATAPI device also supports a limited subset of the ATA command set to facilitate the identification and control of the device. The device supports its primary function (e.g., read or write operations) through a SCSI command set that the device supports. An ATAPI device may implement any command set reportable through the SCSI standard INQUIRY data PERIPHERAL DEVICE TYPE field.

To detect whether an attached device is an ATA device or an ATAPI device, the SATL may issue an ATA IDENTIFY DEVICE command. If the device is an ATAPI device, the device aborts the ATA IDENTIFY DEVICE command and returns a specific signature.

A.3 SCSI CDB transmission

A SCSI CDB is transmitted by a SATL to an ATAPI device by the following sequence:

1) issuing the PACKET command to the device;
2) transmitting the SCSI CDB to the device (the command packet phase); and
3) transmitting or receiving any data, if appropriate, necessary for the completion of the SCSI CDB.

Some ATAPI devices do not permit the Byte Count Limit parameter of the ATA PACKET command to be zero, even if the CDB requires no data transmission. This restriction, or the lack of such a restriction, is specified for each device in ATA IDENTIFY PACKET DEVICE data word 125 (see ATA8-ACS). If the device places a restriction, then the SATL should place a value of 512 in the Byte Count Limit parameter of the ATA PACKET command when transmitted if the CDB being processed requires no data transfer to or from the device.

Within the ATA IDENTIFY PACKET DEVICE data returned by the device is a command size requirement for any SCSI CDB transmitted to the device through the packet command. This restriction is located in word 0 of the ATA IDENTIFY PACKET DEVICE data, bits 1:0. This field specifies either a 12 byte or 16 byte CDB restriction. CDBs of a smaller size may be transmitted by the SATL to the device, however any additional bytes beyond the length of the CDB shall be transmitted as zero.

For example, if an ATAPI device reports that 12 byte command packets are required and the SATL sends an INQUIRY command to the device, the SATL would transmit the six bytes of the INQUIRY command, followed by six bytes set to zero.

If an application client sends a CDB to the SATL, where the size is greater than the maximum command packet size supported by the ATAPI device, then the SATL shall complete the command with CHECK CONDITION status and a sense key of ILLEGAL REQUEST and an additional sense code of INVALID COMMAND OPERATION CODE.

During data transfers, ATAPI devices transmit or receive data on word boundaries. If a SATL transmits a data buffer whose length is not a multiple of a word, then the SATL shall pad the transmitted data with an additional byte set to

zero. During data reception, the SATL shall allocate its receive buffers to accommodate an additional byte if the data length is not a multiple of a word.

In addition to the word alignment requirements, ATAPI devices may have additional requirements imposed on them for padding based on the underlying transport (e.g., SATA ATAPI devices are required to transmit all data aligned to a 32-bit dword. Therefore, a SATL in that environment allocates sufficient receive or transmit buffers to transmit or receive data that has been padded with zeros to a dword boundary).

If the ATAPI device completes a packet command with an error, then the SATL shall send a REQUEST SENSE command to the device through the command transmission mechanism described in this subclause to obtain sense data before completing the CDB to the application client as ATAPI devices do not support any form of autosense. In addition SATL error handling does not use the error translation specified in this standard.

ATAPI devices do not support any form of queued command transmission and SATL implementations shall either maintain an internal queue of received commands for the device or return TASK SET FULL status to the application client if there is already an ATA PACKET command sequence in process or pending for the ATAPI device.

A.4 ATAPI Command management

The PACKET command protocol does not have a mechanism for transmission of task management functions to an ATAPI device. Translation of task management functions by a SATL is unspecified.

A.5 SATL ATAPI implementations

If a SATL supports attachment of ATAPI devices, then the SATL shall not use the translations described elsewhere in this standard for the generation of INQUIRY data and instead shall return the INQUIRY data directly from the ATAPI device. In addition, the SATL shall transmit all SCSI CDBs that are permissible within the command packet data length restrictions (see A.3). The SATL may generate results for the VPD pages 00h and 83h and not send these requests to the ATA device.

A.6 ATAPI I_T nexus loss handling

For an I_T nexus loss event, the SATL shall:

1) issue a software reset to the ATAPI device; and
2) delete all commands in the task set from the SATL internal context.