Seagate®

Product Manual

SCSI Commands Reference Manual

Parallel SCSI

Fibre Channel (FC)

Serial Attached SCSI (SAS)

100293068
Rev. C
April 2010

Publication number: 100293068, Rev. C April 2010

Table of Contents

List of Tables

1.0 Definitions, symbols, abbreviations, keywords, and conventions

This clause contains the definitions, symbols, abbreviations, keywords and, conventions used throughout this manual. These items are from the American National Standards Institute T10 committee standards.

Applicable Documents:

T10/1416-D SCSI Primary Commands - 3 (SPC-3) Revision 23

T10/1731-D SCSI Primary Commands - 4 (SPC-4) Revision 20a

T10/xxxx-D SCSI Block Commands - 3 (SBC-3) Revision 20

T10/1560-D Fibre Channel Protocol for SCSI, Third Edition (FCP-3) Revision 4

T10/1760-D Serial Attached SCSI - 2 (SAS-2) Revision 16

T10/1683-D SCSI Architecture Model - 4 (SAM-4) Revision 14

1.1 Definitions

Access control list (ACL)

The data used by a SCSI target device to configure access rights for initiator ports according to the access controls state of the SCSI target device.

Access control list entry (ACE)

One entry in the access control list.

Access controls

An optional SCSI target device feature that restricts initiator port access to specific logical units and modifies the information about logical units in the parameter data of the INQUIRY and REPORT LUNS commands.

Access controls coordinator

The entity within a SCSI target device that coordinates the management and enforcement of access controls for all logical units within the SCSI target device. The access controls coordinator is always addressable through the ACCESS CONTROLS well known logical unit and LUN 0.

active power condition

When a device server is capable of responding to all of its supported commands, including media access requests, without delay.

additional sense code

A combination of the ADDITIONAL SENSE CODE and ADDITIONAL SENSE CODE QUALIFIER fields in the sense data.

Alias list

A list of alias values and their associated designations maintained by the device server and managed by the CHANGE ALIASES command and REPORT ALIASES command.

Alias value

A numeric value associated to a designation in the alias list and used in command or parameter data to reference a SCSI target device or SCSI target port.

Application client

An object that is the source of SCSI commands. Further definition of an application client may be found in SAM-4.

Attached medium changer

A medium changer that is attached to and accessed through some other type of SCSI device.

Attribute

A single unit of MAM information.

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. A detailed definition of ACA may be found in SAM-4.

Blocked task

A task that is in the blocked state. Tasks become blocked when an ACA condition occurs. The blocked state ends when the ACA condition is cleared. A detailed definition of the blocked task state may be found in SAM-4.

Byte

A sequence of eight contiguous bits considered as a unit.

Cache

See cache memory.

Cache memory

A temporary and often volatile data storage area outside the area accessible by application clients that may contain a subset of the data stored in the non-volatile data storage area.

Check data

Information contained within a redundancy group that may allow lost or destroyed XOR-protected data to be recreated.

Command

A request describing a unit of work to be performed by a device server. A detailed definition of a command may be found in SAM-4.

Command descriptor block (CDB)

The structure used to communicate commands from an application client to a device server. A CDB may have a fixed length of up to 16 bytes or a variable length of between 12 and 260 bytes.

Command standard

A SCSI standard that defines the model, commands, and parameter data for a device type (e.g., SBC-3, SSC-3, SMC-3, MMC-5, or SES-2).

Company_id

Synonym for OUI.

Control mode page

A mode page that provides controls over SCSI features (e.g., task set management and error logging) that are applicable to all device types.

Control Extension mode page

A mode page that provides controls over SCSI features that are applicable to all device types.

Copy manager

The device server that receives an EXTENDED COPY command and performs the operation requested.

Copy target device

The name given by the EXTENDED COPY command to a source or destination logical unit (i.e., a copy target device is a logical unit, not a SCSI target device).

Cyclic redundancy check (CRC)

An error checking mechanism that checks data integrity by computing a polynomial algorithm based checksum.

Data defect list (DLIST)

A list of defects sent by the application client to the device server during a FORMAT UNIT command.

Data-in buffer

The buffer specified by the application client to receive data from the device server during the processing of a command.

Data-out buffer

The buffer specified by the application client to supply data that is sent from the application client to the device server during the processing of a command.

Default protection information

Values placed into protection information fields if an application client does not specify specific protection information values.

Deferred error

A CHECK CONDITION status and sense data that is returned as the result of an error or exception condition that occurred during processing of a previous command for which GOOD, CONDITION MET, INTERMEDIATE, and INTERMEDIATE-CONDITION MET status has already been returned.

Designation

When used in reference to access controls, a name and optional identifier information that specifies a SCSI target device or SCSI target port for association with an alias value in the alias list. Otherwise, a distinguishing name, identifier, or title.

Device Identification VPD page

A VPD page that provides the means to retrieve identification information about the SCSI device, logical unit, and SCSI port.

Device server

An object within a logical unit that processes SCSI tasks according to the rules of task management. A detailed definition of a device server may be found in SAM-4.

Device service request

A request, submitted by an application client, conveying a SCSI command to a device server. A detailed definition of a device service request may be found in SAM-4.

Device service response

The response returned to an application client by a device server on completion of a SCSI command. A detailed definition of a device service response may be found in SAM-4.

Device type

The type of peripheral device (i.e., device model) implemented by the device server and indicated by the contents of the PERIPHERAL DEVICE TYPE field in the standard INQUIRY data.

Direct-access block device

A device that is capable of containing data stored in blocks that each have a unique logical block address.

Disconnect-Reconnect mode page

A mode page that provides the application client the means to tune the performance of the service delivery subsystem.

Domain

An I/O system consisting of a set of SCSI devices that interact with one another by means of a service delivery subsystem.

Element

An addressable physical component of a medium changer SCSI device that may serve as the location of a removable unit of data storage medium. A detailed definition of an element may be found in SMC-3.

Enabled task state

The only task state in which a task may make progress towards completion. A detailed definition of the enabled task state may be found in SAM-4.

Error correcting code (ECC)

An error checking mechanism that checks data integrity and enables some errors in the data to be corrected.

Error history I_T nexus

An I_T nexus for which the device server has reserved access to the error history snapshot (see SPC-4).

Error history snapshot

The contents of the error history at a specific point in time (see SPC-4).

Exclusive-or (XOR)

A Boolean arithmetic function on two binary input values that results in an output value of 1 if one and only one of the input values is 1.

Extended Unique Identifier, a 48-bit globally unique identifier (EUI-48)

The IEEE maintains a tutorial describing EUI-48 at http://standards.ieee.org/regauth/oui/tutorials/EUI48.html.

Extended Unique Identifier, a 64-bit globally unique identifier (EUI-64)

The IEEE maintains a tutorial describing EUI-64 at http://standards.ieee.org/regauth/oui/tutorials/EUI64.html.

Extent

A fixed set of logical blocks occupying contiguous logical block addresses on a single logical unit.

Faulted I_T nexus

The I_T nexus on which a CHECK CONDITION status was returned that resulted in the establishment of an ACA. The faulted I_T nexus condition is cleared when the ACA condition is cleared.

Field

A group of one or more contiguous bits, a part of a larger structure such as a CDB or sense data.

Format corrupt

a vendor-specific condition in which the application client may not be able to perform read operations, write operations, or verify operations.

Grown defect list (GLIST)

All defects sent by the application client to the device server.

Hard reset

A condition resulting from the events defined by SAM-4 in which the SCSI device performs the hard reset operations described in SAM-4, SPC-4, and the applicable command standards.

Host

A SCSI device with the characteristics of a primary computing device, typically a personal computer, workstation, server, minicomputer, mainframe computer, or auxiliary computing device. A host includes one or more SCSI initiator devices.

IEEE company_id

Synonym for OUI.

I_T nexus

A nexus between a SCSI initiator port and a SCSI target port.

I_T nexus loss

A condition resulting from the events defined by SAM-4 in which the SCSI device performs the I_T nexus loss operations described in SAM-4, SPC-4, and the applicable command standards.

I_T_L nexus

A nexus between a SCSI initiator port, a SCSI target port, and a logical unit.

I_T_L_Q nexus transaction

The information transferred between SCSI ports in a single data structure with defined boundaries (e.g., an information unit).

Idle power condition

When a device server is capable of responding to all of its supported commands, including media access requests, but commands may take longer to complete than when in the active power condition.

Implicit head of queue

An optional processing model for specified commands wherein the specified commands may be treated as if they had been received with a HEAD OF QUEUE task attribute.

Initiator device name

A SCSI device name of a SCSI initiator device or of a SCSI target/initiator device when operating as a SCSI initiator device.

Initiator port

Synonymous with SCSI initiator port.

Initiator port identifier

A value by which a SCSI initiator port is referenced within a SCSI domain.

Initiator port name

A SCSI port name of a SCSI initiator port or of a SCSI target/initiator port when operating as a SCSI initiator port.

Internet protocol domain name

The name of a computer or hierarchy of computers within the domain name system defined by the IETF (see RFC 1035 and RFC 1591). The Internet Assigned Numbers Authority maintains a list of domain name assignments at http://www.iana.org/assignments/domain-names.

Internet protocol number

A coded value assigned to identify protocols that layer on the Internet protocol (see RFC 791). The Internet protocol number assigned to the transmission control protocol (TCP, see RFC 793) is six. The Internet Assigned Numbers Authority maintains a list of Internet protocol number assignments at http://www.iana.org/assignments/protocol-numbers.

Linked command

One in a series of SCSI commands processed by a single task that collectively make up a discrete I/O operation. A detailed definition of a linked command may be found in SAM-4.

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).

Left-aligned

A type of field containing ASCII data in which unused bytes are placed at the end of the field (highest offset) and are filled with ASCII space (20h) characters.

Logical block

A set of data bytes accessed and referenced as a unit.

Logical block address (LBA)

The value used to reference a logical block.

Logical unit

An externally addressable entity within a SCSI target device that implements a SCSI device model and contains a device server. A detailed definition of a logical unit may be found in SAM-4.

Logical unit access control descriptor (LUACD)

The structure within an ACE that identifies a logical unit to which access is allowed and specifies the LUN by which the logical unit is to be accessed.

Logical unit certification list (CLIST)

Defects detected by the device server during an optional certification process performed during the FORMAT UNIT command.

Logical unit inventory

The list of the logical unit numbers reported by a REPORT LUNS command.

Logical unit number (LUN)

An encoded 64-bit identifier for a logical unit. A detailed definition of a logical unit number may be found in SAM-4.

Logical unit reset

A condition resulting from the events defined by SAM-4 in which the logical unit performs the logical unit reset operations described in SAM-4, SPC-4, and the applicable command standards.

Media

Plural of Medium

Medium

A physical entity that stores data in a nonvolatile manner (i.e., retained through a power cycle) in accordance with commands processed by the device server.

Medium auxiliary memory (MAM)

An auxiliary memory residing on a medium that is accessible to the device server (e.g., a tape cartridge). Medium auxiliary memory may be nonvolatile and independent of the main function of the device server.

Medium changer

A device that mechanizes the movement of media to and from the SCSI device that records on or reads from the media. A detailed definition of a medium changer may be found in SMC-3.

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).

Name

A label of an object that is unique within a specified context and should never change (e.g., the term name and worldwide identifier (WWID) may be interchangeable).

Network address authority (NAA)

A field within a name that specifies the format and length of that name. See FC-FS.

Nexus

A relationship between two SCSI devices, and the SCSI initiator port and SCSI target port objects within those SCSI devices.

Non-volatile cache

Cache that retains data through power cycles.

Non-volatile cache memory

Cache memory that retains data through power cycles.

Non-volatile medium

A physical storage medium that retains data written to it for subsequent read operations through power cycles (e.g., a disk within a device that stores data as magnetic field changes that do not require device power to exist).

Null-padded

A type of field in which unused bytes are placed at the end of the field (i.e., highest offset) and are filled with ASCII null (00h) characters.

Null-terminated

A type of field in which the last used byte (i.e., highest offset) is required to contain an ASCII null (00h) character.

One

The logical true condition of a variable.

Operation Code

The first byte of a SCSI CDB shall contain an operation code identifying the operation being requested by the CDB.

Organizationally unique identifier (OUI)

A numeric identifier that is assigned by the IEEE such that no assigned identifiers are identical. OUI is equivalent to company_id or IEEE company_id. The IEEE prefers OUI for EUI-48 identifiers and company_id for EUI-64 identifiers. However, the numeric identifier is called an OUI when it is assigned by the IEEE. The IEEE maintains a tutorial describing the OUI at http://standards.ieee.org/regauth/oui/.

Page

A regular parameter structure (or format) used by several commands. These pages are identified with a value known as a page code.

Persist through power loss

An optional capability associated with some features that allows an application client to request that a device server maintain information regarding that feature across power failures.

Persistent reservation holder

The I_T nexus(es) that are allowed to release or change a persistent reservation without preempting it.

Power cycle

Power being removed from and later applied to a SCSI device.

Power on

A condition resulting from the events defined by SAM-4 in which the SCSI device performs the power on operations described in SAM-4, SPC-4, and the applicable command standards.

Primary defect list (PLIST)

The list of defects that are considered permanent defects.

Protection information

Fields appended to each logical block that contain a cyclic redundancy check (CRC), an application tag, and a reference tag.

Protocol identifier

A coded value used in various fields to identify the protocol to which other fields apply.

Protocol specific

A requirement that is defined by a SCSI transport protocol standard. A detailed definition of protocol specific may be found in SAM-4.

Protocol standard

A SCSI standard that defines SCSI transport protocol (e.g., SAS, SPI-5, SBP-3, or FCP-2).

Proxy token

An identifier for a logical unit that may be used to gain temporary access to that logical unit in the presence of access controls.

Redundancy group

A grouping of XOR-protected data and associated check data into a single type of data redundancy (see SCC-2). SPC-4 only supports the XOR type of redundancy.

Request for comment (RFC)

The name given to standards developed by the Internet Engineering Task Force.

Registered

The condition that exists for an I_T nexus following the successful completion of a PERSISTENT RESERVE OUT command with a REGISTER service action, REGISTER AND IGNORE EXISTING KEY service action, or REGISTER AND MOVE service action and lasting until the registration is removed.

Registrant

An I_T nexus that is registered.

Right-aligned

A type of field containing ASCII data in which unused bytes are placed at the start of the field (i.e., lowest offset) and are filled with ASCII space (20h) characters.

Relative port identifier

An identifier for a SCSI port that is unique within a SCSI device. Application clients may use the SCSI Ports VPD page to determine relative port identifier values.

Relative initiator port identifier

A relative port identifier for a SCSI initiator port.

Relative target port identifier

A relative port identifier for a SCSI target port.

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.

SCSI device name

A name of a SCSI device that is world wide unique within the protocol of a SCSI domain in which the SCSI device has SCSI ports. The SCSI device name may be made available to other SCSI devices or SCSI ports in protocol specific ways.

SCSI domain

The interconnection of two or more SCSI devices and a service delivery subsystem. A detailed definition of a SCSI Domain may be found in SAM-4.

SCSI initiator device

A SCSI device containing application clients and SCSI initiator ports that originate device service and task management requests to be processed by a SCSI target device and receives device service and task management responses from SCSI target devices.

SCSI initiator port

A SCSI initiator device object acts as the connection between application clients and the service delivery subsystem through which requests and responses are routed.

SCSI port

A port of a SCSI device that connects the application client, device server or task manager to the service delivery subsystem.

SCSI port identifier

A value by which a SCSI port is referenced within a domain. The SCSI port identifier is either an initiator port identifier or a target port identifier.

SCSI port name

A name of a SCSI port that is world wide unique within the protocol of the SCSI domain of that SCSI port. The name may be made available to other SCSI devices or SCSI ports in that SCSI domain in protocol specific ways.

SCSI Ports VPD page

A VPD page that allows retrieval of information about all the SCSI ports in a SCSI target device or SCSI target/initiator device.

SCSI target device

A SCSI device containing logical units and SCSI target ports that receives device service and task management requests for processing and sends device service and task management responses to SCSI initiator devices.

SCSI target port

A SCSI target device object that acts as the connection between device servers and task managers and the service delivery subsystem through which requests and responses are routed.

SCSI transport protocol standard

A SCSI standard that defines a SCSI transport protocol (e.g., FCP-2, SAS, SRP, or SBP-3).

Sense data

Data describing an error or exceptional condition that a device server delivers to an application client in the same I_T_L_Q nexus transaction as a CHECK CONDITION status or in response to a REQUEST SENSE command. The format of sense data is defined in SPC-4.

Sense key

The contents of the SENSE KEY field in the sense data.

Service action

A request describing a unit of work to be performed by a device server. A service action is an extension of a command.

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.

Standby power condition

When a device server is capable of accepting commands, but not capable of processing media access commands.

Status

One byte of response information sent from a device server to an application client upon completion of each command.

Storage array controller

Any combination of an initiator and application clients (see SAM-4) that originates SCSI commands, converts input LUNs to output LUNs, and converts input LBAs to output LBAs. A storage array controller organizes a group of direct-access block devices into various objects (e.g., redundancy groups and volume sets). See SCC-2.

System

One or more SCSI domains operating as a single configuration.

Target device name

A SCSI device name of a SCSI target device or of a SCSI target/initiator device when operating as a SCSI target device.

Target port

Synonymous with SCSI target port.

Target port asymmetric access state

The characteristic that defines the behavior of a target port and the allowable command set for a logical unit when commands and task management functions are routed through the target port maintaining that state.

Target port group

A set of target ports that are in the same target port asymmetric access state at all times.

Target port group asymmetric access state

The target port asymmetric access state common to the set of target ports in a target port group.

Target port identifier

A value by which a SCSI target port is referenced within a SCSI domain.

Target port name

A SCSI port name of a SCSI target port or of a SCSI target/initiator port when operating as a SCSI target port.

Task

An object within a logical unit that represents the work associated with a command or a group of linked commands. A detailed definition of a task may be found in SAM-4.

Task set

A group of tasks within a logical unit, whose interaction is dependent on the task management (queuing) and ACA rules. See SAM-4 and the Control mode page.

TCP port numbers

One of the data needed to establish a TCP connection. TCP port numbers may be assigned to protocols that layer on TCP by the Internet Assigned Numbers Authority. The Internet Assigned Numbers Authority maintains a list of TCP port number assignments at http://www.iana.org/assignments/ port-numbers.

Third-party command

A command sent to one SCSI device requesting than an operation be performed involving two other SCSI devices (e.g., the EXTENDED COPY command may perform copy operations between two or more SCSI devices none of which are the SCSI device to which the EXTENDED COPY command was sent).

Unit attention condition

A state that a logical unit maintains while it has asynchronous status information to report to the initiator ports associated with one or more I_T nexuses. See SAM-4.

Universal time (UT)

The time at longitude zero, colloquially known as Greenwich Mean Time. See http://aa.usno.navy.mil/faq/docs/UT.html.

URI Schemes

The Internet Assigned Numbers Authority maintains a list of schemes for URI and URL names at http://www.iana.org/assignments/uri-schemes.

User data

Data contained in logical blocks that is not protection information.

UTF-8

A character set that is a transformation format of the character set defined by ISO 10646. See RFC 2279.

Vendor specific (VS)

Something (e.g., a bit, field, or code value) that is not defined by SPC-4 and may be vendor defined.

Volatile cache memory or Volatile cache

Cache memory that does not retain data through power cycles.

Volatile medium

Medium that does not retain data written to it for a subsequent read operation through power cycles (e.g., a silicon memory device that loses data written to it if device power is lost).

Well known logical unit

A logical unit that only does specific functions. Well known logical units allow an application client to issue requests to receive and manage specific information usually relating to a SCSI target device.

Well known logical unit number (W-LUN)

The logical unit number that identifies a well known logical unit.

XOR operation

Performing an XOR bitwise on two identical-sized multiple-bit input values (e.g., the current value of a logical block and the new value for that logical block). In a storage array implementing a redundancy group, the XOR operation is used in error correction algorithms and may be performed by the storage array controller or by the direct-access block devices.

XOR-protected data

Logical blocks, including user data and protection information, if any, that are part of a redundancy group.

Zero

The logical false condition of a variable.

Zero-padded

A type of field in which unused bytes are placed at the end of the field (i.e., highest offset) and are filled with zeros.

1.2 Symbols and abbreviations

See Table 1 for abbreviations of standards bodies (e.g., ISO). Additional symbols and abbreviations used in the manual include:

Abbreviation	Meaning
<	less than
>	greater than
ACE	Access Control list Entry
ACL	Access Control List
ACA	Auto Contingent Allegiance
ADC	Automation/Drive Interface - Commands
ADT	Automation/Drive Interface - Transport Protocol
ASC	Additional Sense Code
ASCII	American Standard Code for Information Interchange
ASCQ	Additional Sense Code Qualifier
ATA	AT Attachment (see www.t13.org)
ATAPI	AT Attachment with Packet Interface (see www.t13.org)
CDB	Command Descriptor Block
CRC	Cyclic Redundancy Check
CLIST	logical unit certification list
D_ID	Destination Identifier (defined in FC-FS)
DLIST	data defect list
ECC	error correcting code
EUI-48	Extended Unique Identifier, a 48-bit globally unique identifier
EUI-64	Extended Unique Identifier, a 64-bit globally unique identifier
FC-FS	Fibre Channel Framing and Signaling Interface
FCP-2	Fibre Channel Protocol for SCSI -2
GLIST	grown defect list
HTTP	Hypertext Transfer Protocol (see RFC 2616)
I/O	input/output
ID	Identifier or Identification
IEC	International Electrotechnical Commission

IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
iSCSI	Internet SCSI
ISO	Organization for International Standards
LBA	Logical Block Address
LSB	Least Significant Bit
LUACD	Logical Unit Access Control Descriptor
LUN	Logical Unit Number
MAM	Medium Auxiliary Memory
MMC-5	SCSI Multi-Media Commands -5
MSB	Most Significant Bit
NAA	Network Address Authority
n/a	not applicable
INCITS	InterNational Committee for Information Technology Standards
OCRW	SCSI Specification for Optical Card Reader/Writer
OSD	Object-based Storage Devices Commands
OUI	Organizationally Unique Identifier
PLIST	primary defect list
RAID	Redundant Array of Independent Disks
RBC	SCSI Reduced Block Commands
RDMA	Remote Direct Memory Access (see SRP)
RFC	Request For Comments
RMC	SCSI Reduced Multi-Media Commands
SAM-2	SCSI Architecture Model -2
SAM-3	SCSI Architecture Model -3
SAM-4	SCSI Architecture Model -4
SAT	SCSI / ATA Translation
SBC-2	SCSI Block Commands -2
SBC-3	SCSI Block Commands -3
SBP-3	Serial Bus Protocol -3
SCC-2	SCSI Controller Commands -2
SCC-3	SCSI Controller Commands -3
SCSI	The architecture defined by the family of standards described in clause 1
SES	SCSI-3 Enclosure Services
SES-2	SCSI Enclosure Services -2
SMC-2	SCSI Media Changer Commands -2
SMC-3	SCSI Media Changer Commands -3
SPC	SCSI-3 Primary Commands (ANSI INCITS 301-1997)
SPC-2	SCSI Primary Commands -2
SPC-3	SCSI Primary Commands -3
SPC-4	SCSI Primary Commands -4
SPI-5	SCSI Parallel Interface -5
SRP	SCSI RDMA Protocol
SSC-3	SCSI Stream Commands -2

TCP	Transmission Control Protocol (see RFC 793)
URI	Uniform Resource Identifier (see RFC 2396 and RFC 3305)
URL	Uniform Resource Locator (see RFC 2396 and RFC 3305)
UT	Universal time
USB	Universal Serial Bus (see www.usb.org)
VPD	Vital Product Data
VS	Vendor Specific
W-LUN	Well known logical unit number

1.3 Keywords

Expected

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

Ignored

A keyword used to describe an unused bit, byte, word, field or code value. The contents or value of an ignored bit, byte, word, field or code value shall not be examined by the receiving SCSI device and may be set to any value by the transmitting SCSI device.

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.

Mandatory

A keyword indicating an item that is required to be implemented as defined in this manual.

May

A keyword that indicates flexibility of choice with no implied preference (equivalent to “may or may not”).

May not

Keywords that indicate flexibility of choice with no implied preference (equivalent to “may or may not”).

Need not

Keywords indicating a feature that is not required to be implemented (equivalent to “is not required to”).

Obsolete

A keyword indicating that an item was defined in prior SCSI standards but has been removed from this manual.

Optional

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

Reserved

A keyword referring to bits, bytes, words, fields and code values that are set aside for future standardization. A reserved bit, byte, word or field shall be set to zero, or in accordance with a future extension to this manual. 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.

Restricted

A keyword referring to bits, bytes, words, and fields that are set aside for use in other SCSI standards. A restricted bit, byte, word, or field shall be treated as a reserved bit, byte, word or field for the purposes of the requirements defined in this manual.

Shall

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

Should

A keyword indicating flexibility of choice with a strongly preferred alternative; equivalent to the phrase “it is strongly recommended.”

Vendor-specific

Something (e.g., a bit, field, or code value) that is not defined by this manual and may be used differently in various implementations.

1.4 Conventions

Certain words and terms used in this manual have a specific meaning beyond the normal English meaning. These words and terms are defined either in this clause or in the text where they first appear. Names of commands, status codes, sense keys, and additional sense codes are in all uppercase (e.g., REQUEST SENSE).

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.

Names of fields and state variables are in uppercase (e.g. NAME). When a field or state variable name contains acronyms, uppercase letters may be used for readability. Normal case is used when the contents of a field or state variable are being discussed. Fields or state variables containing only one bit are usually referred to as the NAME bit instead of the NAME field.

Normal case is used for words having the normal English meaning.

A binary number is represented in this manual by any sequence of digits comprised 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 between characters 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 manual by any sequence of digits comprised 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 FD8CFA23h or B_FD8C_FA23h).

A decimal number is represented in this manual by any sequence of digits comprised 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 manual uses the ISO convention for representing decimal numbers (e.g., the thousands and higher multiples are separated by a space and a comma is used as the decimal point). Table 1 shows some examples of decimal numbers represented using the ISO and American conventions.

Table 1. ISO v American Numbering Conventions

ISO	American
0,6	0.6
3,141 592 65	3.14159265
1 000	1,000
1 323 462	1,323,462.95

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

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

2.0 General Concepts

This manual defines behaviors that are common to all Seagate SCSI device models. This manual defines the SCSI commands that are basic to more than one disc drive model and the SCSI commands that may apply to any SCSI Interface, including Parallel, Fibre Channel, and Serial Attached SCSI (SAS).

2.1 Command Descriptor Block (CDB)

2.1.1 CDB usage and structure

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

If a logical unit validates reserved CDB fields and receives a reserved field within the CDB that is not zero, then the logical unit 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 a logical unit receives a reserved CDB code value in a field other than the OPERATION CODE field, then the logical unit 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 fixed length CDB formats are described in 2.1.2. The variable length CDB formats are described in 2.1.4. The CDB fields that are common to most commands are described in 2.1.5. The fields shown in 2.1.2 and 2.1.3 and described in 2.1.4 are used consistently by most commands. However, the actual usage of any field (except OPERATION CODE and CONTROL) is described in the subclause defining that command. If a device server receives a CDB containing an operation code that is invalid or not supported, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

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

2.1.2 The fixed length CDB formats

All fixed length CDBs shall have an OPERATION CODE field as their first byte and a CONTROL byte as their last byte. Table 2 shows the typical format of a 6-byte CDB. Table 3 shows the typical format of a 10-byte CDB. Table 4 shows the typical format of a 12-byte CDB. Table 5 shows the typical format of a 16-byte CDB. Table 6 shows the format of a 16-byte CDB for commands that provide for a long LBA.

Table 2. Typical CDB for 6-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			(MSB)				
2	LOGICAL BLOCK ADDRESS (if required)							
3								(LSB)
4	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
5	CONTROL							

Table 3. Typical CDB for 10-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								(LSB)
6	Miscellaneous CDB information							
7	(MSB)	TRANSFER LENGTH (if required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
8								(LSB)
9	CONTROL							

Table 4. Typical CDB for 12-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)				
2	(MSB)	LOGICAL BLOCK ADDRESS (if required)						
3								
4								
5								(LSB)
6	(MSB)	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
7								
8								
9								(LSB)
10	Miscellaneous CDB information							
11	CONTROL							

Table 5. Typical CDB for 16-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information			SERVICE ACTION (if required)]				
2	(MSB)							
3	LOGICAL BLOCK ADDRESS (if required)							
4								
5								(LSB)
6	(MSB)							
7	Additional CDB data (if required)							
8								
9								(LSB)
10	(MSB)							
11	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)							
12								
13								(LSB)
14	Miscellaneous CDB information							
15	CONTROL							

Table 6. Typical CDB for long LBA 16-byte commands

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Miscellaneous CDB information							
2	(MSB)	LOGICAL BLOCK ADDRESS						
3								
4								
5								
6								
7								
8								
9								(LSB)
10	(MSB)	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
11								
12								
13								(LSB)
14	Miscellaneous CDB information							
15	Control							

2.1.3 The variable length CDB formats

The first byte of a variable length CDB shall contain the operation code 7Fh. The CONTROL byte is the second byte in the variable length CDB (see table 7)

Table 7. Typical variable length CDB

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Miscellaneous CDB information							
3	Miscellaneous CDB information							
4	Miscellaneous CDB information							
5	Miscellaneous CDB information							
6	Miscellaneous CDB information							
7	ADDITIONAL CDB LENGTH (n–7)							
8	(MSB)	SERVICE ACTION						
9								(LSB)
10 : n	Service Action specific fields							

ADDITIONAL CDB LENGTH field

The ADDITIONAL CDB LENGTH field specifies the number of additional CDB bytes. This value in the ADDITIONAL CDB LENGTH field shall be a multiple of 4. If the number of CDB bytes delivered by the service delivery subsystem is not sufficient to contain the number of bytes specified by the ADDITIONAL CDB LENGTH field, then the 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.

SERVICE ACTION field

The SERVICE ACTION field specifies the action being requested by the application client. The SERVICE ACTION field is required in the variable length CDB format and is described in 4.3.4.2. Each service action code description defines a number of service action specific fields that are needed for that service action.

A 32-byte variable length CDB format is defined for long LBA operations (see table 8)

Table 8. Typical variable length CDB for long LBA 32-byte commands

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Miscellaneous CDB information							
3	Miscellaneous CDB information							
4	Miscellaneous CDB information							
5	Miscellaneous CDB information							
6	Miscellaneous CDB information							
7	Additional CDB Length (n–7) [9]							
8	(MSB)	SERVICE ACTION						
9								(LSB)
10	Miscellaneous CDB information			DPO	FUA	Miscellaneous CDB information		
11	Miscellaneous CDB information							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20 27	Miscellaneous CDB information							
28	(MSB)	TRANSFER LENGTH (If required) PARAMETER LIST LENGTH (if required) ALLOCATION LENGTH (if required)						
31								(LSB)

2.2 Common CDB fields

2.2.1 Operation Code

The first byte of a SCSI CDB shall contain an operation code identifying the operation being requested by the CDB. Some operation codes provide for modification of their operation based on a service action (see 2.1.4.2). In such cases, the operation code and service action code combine to identify the operation being requested. The location of the SERVICE ACTION field in the CDB varies depending on the operation code value.

The OPERATION CODE (see table 10) of the CDB has a GROUP CODE field and a COMMAND CODE field. The three-bit GROUP CODE field provides for eight groups of command codes. The five-bit COMMAND CODE field provides for thirty-two command codes in each group. A total of 256 possible operation codes exist. Operation codes are defined in this manual and other command standards. The group code value shall determine the length of the CDB (see table 11).

Table 9. OPERATION CODE byte

Bit	7	6	5	4	3	2	1	0
	GROUP CODE			COMMAND CODE				

The value in the GROUP CODE field specifies one of the groups shown in Table 10.

Table 10. Group Code values

Group Code	Meaning	Typical CDB Format
000b	6 byte commands	see Table 2
001b	10 byte commands	see Table 3
010b	10 byte commands	see Table 3
011b	Reserved [a]	
100b	16 byte commands	see Table 5 and Table 6
101b	12 byte commands	see Table 4
110b	Vendor Specific	
111b	Vendor Specific	

[a] The format of the commands using operation code 7Fh is described in 2.1.3. With the exception of operation code 7Fh, all group code 011b operation codes are reserved.

2.2.2 SERVICE ACTION

All CDB formats except the 6-byte format provide for a SERVICE ACTION field containing a coded value identifying a function to be performed under the more general command function specified in the OPERATION CODE field. While the SERVICE ACTION field is defined for CDB formats, it is used as described in this subclause only in those CDB formats that contain a SERVICE ACTION field. When the specific field SERVICE ACTION is not defined in a CDB format, the bits identified as the SERVICE ACTION field in a CDB shall be used or reserved as specified by the particular CDB format.

2.2.3 Logical block address

The logical block addresses on a logical unit or within a volume or partition shall begin with block zero and be contiguous up to the last logical block of that logical unit or within that volume or partition.

A six-byte CDB may contain a 21-bit LOGICAL BLOCK ADDRESS field. The ten-byte and the twelve-byte CDBs may contain 32-bit LOGICAL BLOCK ADDRESS fields. The sixteen-byte CDB has two formats: one allows a 32-bit LOGICAL BLOCK ADDRESS field (see Table 5) and the other allows a 64-bit LOGICAL BLOCK ADDRESS field (see Table 6). LOGICAL BLOCK ADDRESS fields in additional parameter data have their length specified for each occurrence. See the specific command descriptions.

2.2.4 TRANSFER LENGTH

The TRANSFER LENGTH field specifies the amount of data to be transferred, usually the number of blocks. Some commands use transfer length to specify the requested number of bytes to be sent as defined in the command description.

Commands that use one byte for the TRANSFER LENGTH field may allow up to 256 blocks or 256 bytes of data to be transferred by one command.

In commands that use multiple bytes for the TRANSFER LENGTH field, a transfer length of zero specifies that no data transfer shall take place. A value of one or greater specifies the number of blocks or bytes that shall be transferred. Refer to the specific command description for further information.

2.2.5 PARAMETER LIST LENGTH

The PARAMETER LIST LENGTH field is used to specify the number of bytes sent from the Data-Out Buffer. This field is typically used in CDBs for parameters that are sent to a device server (e.g., mode parameters, diagnostic parameters, log parameters). A parameter list length of zero specifies that no data shall be transferred. This condition shall not be considered as an error, unless otherwise specified.

2.2.6 ALLOCATION LENGTH

The ALLOCATION LENGTH field specifies the maximum number of bytes that an application client has allocated in the Data-In Buffer. An allocation length of zero specifies that no data shall be transferred. This condition shall not be considered as an error. The device server shall terminate transfers to the Data-In Buffer when the number of bytes specified by the ALLOCATION LENGTH field have been transferred or when all available data have been transferred, whichever is less. The allocation length is used to limit the maximum amount of variable length data (e.g., mode data, log data, diagnostic data) returned to an application client. If the information being transferred to the Data-In Buffer includes fields containing counts of the number of bytes in some or all of the data, then the contents of these fields shall not be altered to reflect the truncation, if any, that results from an insufficient ALLOCATION LENGTH value, unless this manual describes the Data-In Buffer format states otherwise.

If the amount of information to be transferred exceeds the maximum value that the ALLOCATION LENGTH field is capable of specifying, the device server shall transfer no data and 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.

2.2.7 CONTROL

The CONTROL byte has the same definition for all commands.

All CDBs shall contain a CONTROL byte (see table 11). The location of the CONTROL byte within a CDB depends on the CDB format (see “<$T and “<$T).

Table 11. Control Byte

Bit	7	6	5	4	3	2	1	0
	Vendor Specific		Reserved			NACA	Obsolete [1]	Obsolete [2]

[1] Bit 1 of the Control byte was formerly the FLAG bit. See FLAG bit below).

[2] Bit 0 of the Control byte was formerly the LINK bit. See LINK bit below).

All SCSI transport protocol standards shall define the functionality needed for a logical unit to implement the NACA bit and LINK bit.

NACA (Normal ACA) bit

The NACA (Normal ACA) bit specifies whether an auto contingent allegiance (ACA) is established if the command returns with CHECK CONDITION status. An NACA bit set to one specifies that an ACA shall be established.

0 An NACA bit set to zero specifies that an ACA shall not be established. The actions for ACA are specified in SAM-4. Actions that may be required when an ACA is not established are described in SAM-4. All logical units shall implement support for the NACA value of zero and may support the NACA value of one (i.e., ACA). The ability to support a NACA value of one is indicated with the NORMACA bit in the standard INQUIRY data (see clause 3.6.2)

1 If the NACA bit is set to one but the logical unit does not support ACA, the 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.

LINK bit -- declared Obsolete by T10

The LINK bit is used to continue the task across multiple commands. Support for the LINK bit is optional. The application client sets the LINK bit to one to specify a request for continuation of the task across two or more commands.

1 If the LINK bit is set to one and the command completes successfully, a logical unit that supports the LINK bit shall continue the task and return a status of INTERMEDIATE or INTERMEDIATE-CONDITION MET and a service response of LINKED COMMAND COMPLETE (see SAM-3). If the LINK bit is set to one and the logical unit does not support linked commands, the 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.D FIELD IN CDB.

FLAG bit -- declared Obsolete by T10

Note. In older model drives the FLAG bit may have been used in conjunction with the LINK bit to notify the initiator in an expedited manner that the command has completed.

Support for the FLAG bit is a logical unit option. If the LINK bit and FLAG bit are both set to one, and if the command completes with a status of Intermediate or Intermediate-Condition Met, a drive that supports the FLAG bit returns a service response of Linked Command Complete (with Flag).

The drive completes the command with a status of Check Condition and a sense key of Illegal Request if:

[a] • The LINK bit is set to one and the drive does not support linked commands, or

[b] • The FLAG bit is set to one and the drive does not support the FLAG bit, or

[c] • The FLAG bit is set to one and the LINK bit is set to zero.

2.2.8 Grouping function

A grouping function is a function that collects information about attributes associated with commands (i.e., information about commands with the same group value are collected into the specified group). The definition of the attributes and the groups is outside the scope of this manual. Groups are identified with the GROUP NUMBER field in the CDB of certain commands (e.g., the WRITE (10) command (see 3.62)).

The collection of this information is outside the scope of this manual (e.g., the information may not be transmitted using any SCSI protocols).

Note. An example of how grouping could be used, consider two applications using a subsystem; one application streams data and another accesses data randomly. If the streaming application groups all of its commands with one value (e.g., x), and the random application groups all of its commands with another value (e.g., y), then a group x defined to hold performance metrics collects all the performance metrics for the streamed commands together and a group y defined to also hold performance metrics collects all the performance metrics for the random commands together. The result is two sets of performance metrics (i.e., x and y). A management application then reads the performance metrics and determines if the performance of a specific group is acceptable.

Support for the grouping function is indicated in the GROUP_SUP bit in the Extended INQUIRY Data VPD page (see 4.4.6).

2.3 Parameter rounding

Certain parameters sent to a SCSI target port with various commands contain a range of values. Targets may choose to implement only selected values from this range. When the target receives a value that it does not support, it either rejects the command (CHECK CONDITION status with Illegal Request Sense key) or it rounds the value received to a supported value. The target shall reject unsupported values unless rounding is permitted in the description of the parameter.

Rounding of parameter values, when permitted (Rounding is enabled by MODE SELECT command, page code 00h, byte 2, bit 2) shall be performed as follows:

a SCSI target port that receives a parameter value that is not an exact supported value shall adjust the value to one that it supports and shall return CHECK CONDITION status with a sense key of Recovered Error. The additional sense code shall be set to Rounded Parameter. The initiator is responsible to issue an appropriate command to learn what value the target has selected.

Implementor’s Note: Generally, the target should adjust maximum-value fields down to the next lower supported value than the one specified by the initiator. Minimum-value fields should be rounded up to the next higher supported value than the one specified by the initiator. In some cases, the type of rounding (up or down) is explicitly specified in the description of the parameter.

2.4 Sense data

2.4.1 Sense data introduction

Sense data shall be returned in the same I_T_L_Q nexus transaction as a CHECK CONDITION status and as parameter data in response to the REQUEST SENSE command. Sense data returned in the same I_T_L_Q nexus transaction as a CHECK CONDITION status shall be either fixed or descriptor format sense data format based on the value of the D_SENSE bit in the Control mode page. The REQUEST SENSE command may be used to request either the fixed format sense data or the descriptor format sense data.f

The first byte of all sense data contains the RESPONSE CODE field that indicates the error type and format of the sense data (see table 12).

Table 12. Sense data response codes

Response Code	Error type		Sense data format	
	Description	**Reference**	**Description**	**Reference**
00h - 6Fh	Reserved			
70h	Current	2.4.1.3	Fixed	2.4.1.2
71h	Deferred	2.4.1.4	Fixed	2.4.1.2
72h	Current	2.4.1.3	Descriptor	2.4.1.1
73h	Deferred	2.4.1.4	Descriptor	2.4.1.1
74h - 7Eh	Reserved			
7Fh	Vendor specific			

The RESPONSE CODE field shall be set to 70h in all unit attention sense data in which:

a) The ADDITIONAL SENSE CODE field is set to 29h; or

b) The additional sense code is set to MODE PARAMETERS CHANGED.

2.4.1.1 Descriptor format sense data

2.4.1.1.1 Descriptor format sense data overview

The descriptor format sense data for response codes 72h (current errors) and 73h (deferred errors) is defined in table 13.

Table 13. Descriptor format sense data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved	RESPONSE CODE (72h or 73h)						
1	Reserved				SENSE KEY			
2	ADDITIONAL SENSE CODE							
3	ADDITIONAL SENSE CODE QUALIFIER							
4	Reserved							
5								
6								
7	ADDITIONAL SENSE LENGTH (n-7)							
Sense Data Descriptor(s)								
8	SENSE DATA DESCRIPTOR 0 (see table 14)							
	. . .							
	SENSE DATA DESCRIPTOR X (see table 14)							
n								

RESPONSE CODE field

The contents of the RESPONSE CODE field indicate the error type and format of the sense data (see 2.4.1). For descriptor format sense data, the RESPONSE CODE field shall be set to 72h or 73h.

SENSE KEY, ADDITIONAL SENSE CODE and ADDITIONAL SENSE CODE QUALIFIER fields

The SENSE KEY, ADDITIONAL SENSE CODE and ADDITIONAL SENSE CODE QUALIFIER fields provide a hierarchy of information. The hierarchy provides a top-down approach for an application client to determine information relating to the error and exception conditions.

SENSE KEY field

The SENSE KEY field indicates generic information describing an error or exception condition. The sense keys are defined in 2.4.1.5.

ASC (ADDITIONAL SENSE CODE) field

The ADDITIONAL SENSE CODE (ASC) field indicates further information related to the error or exception condition reported in the SENSE KEY field. Support of the additional sense codes not required by this manual is optional. A list of additional sense codes is in 2.4.1.5. If the device server does not have further information related to the error or exception condition, the additional sense code shall be set to zero.

ASCQ (ADDITIONAL SENSE CODE QUALIFIER) field

The ADDITIONAL SENSE CODE QUALIFIER (ASCQ) field indicates detailed information related to the additional sense code. If the error or exception condition is reported by the device server, the value returned shall be as specified in 2.4.1.5. If the device server does not have detailed information related to the error or exception condition, the additional sense code qualifier shall be set to zero.

ADDITIONAL SENSE LENGTH field

The ADDITIONAL SENSE LENGTH field indicates the number of additional sense bytes that follow. The additional sense length shall be less than or equal to 244 (i.e., limiting the total length of the sense data to 252 bytes). If the sense data is being returned as parameter data by a REQUEST SENSE command, then the relationship between the ADDITIONAL SENSE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

SENSE DATA DESCRIPTORS field

Sense data descriptors (see table 14) provide specific sense information. A given type of sense data descriptor shall be included in the sense data only when the information it contains is valid.

Table 14. Sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE							
1	ADDITIONAL LENGTH (n-1)							
2	SENSE DATA DESCRIPTOR SPECIFIC							
n								

DESCRIPTOR TYPE field

The DESCRIPTOR TYPE field contains a type code (see table 15) that identifies the type of sense data descriptor. No more than one sense data descriptor of each type shall be included in the descriptor format sense data.

Table 15. Sense data descriptor types

Type	Description	Reference
00h	Information	2.4.1.1.2
01h	Command specific information	2.4.1.1.3
02h	Sense key specific	2.4.1.1.4
03h	Field replaceable unit	2.4.1.1.5
04h	Stream commands	SSC-3
05h	Block commands	SBC-3
06h	OSD object identification	OSD
07h	OSD response integrity check value	OSD
08h	OSD attribute identification	OSD
09h	ATA Return	SAT
0Ah	Progress indication	2.4.1.1.6
0Bh - 7Fh	Reserved	
80h - FFh	Vendor specific	2.4.1.1.7

ADDITIONAL LENGTH field

The ADDITIONAL LENGTH field indicates the number of sense data descriptor specific bytes that follow in the sense data descriptor.

2.4.1.1.2 Information sense data descriptor

The information sense data descriptor (see table 16) provides information that is device-type or command specific and is defined in a command standard.

Table 16. Information sense data descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (00h)							
1	ADDITIONAL LENGTH (0Ah)							
2	VALID (1b)	Reserved						
3	Reserved							
4	INFORMATION							
11								

DESCRIPTOR TYPE and ADDITIONAL LENGTH fields

The DESCRIPTOR TYPE and ADDITIONAL LENGTH fields are described in 2.4.1.1.1. For the information sense data descriptor, the DESCRIPTOR TYPE field shall be set to 00h and the ADDITIONAL LENGTH field shall be set to 0Ah.

VALID bit

The VALID bit shall be set to one.

Note. In the fixed format sense data, the VALID bit indicates whether the contents of the INFORMATION field is valid as defined by a command standard. Since the contents of the INFORMATION field are valid whenever an information sense data descriptor is included in the sense data, the only legal value for the VALID bit is set to one.

INFORMATION field

The contents of the INFORMATION field are device-type or command specific and are defined in a command standard. When a four byte quantity is stored in the INFORMATION field, the first four bytes shall be zero.

2.4.1.1.3 Command-specific information sense data descriptor

The command-specific information sense data descriptor (see table 17) provides information that depends on the command on which the exception condition occurred.

Table 17. Command-specific information sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (01h)							
1	ADDITIONAL LENGTH (0Ah)							
2	Reserved							
3	Reserved							
4	COMMAND-SPECIFIC INFORMATION							
11								

DESCRIPTOR TYPE and ADDITIONAL LENGTH fields

The DESCRIPTOR TYPE and ADDITIONAL LENGTH fields are described in 2.4.1.1.1. For the command-specific information sense data descriptor, the DESCRIPTOR TYPE field shall be set to 01h and the ADDITIONAL LENGTH field shall be set to 0Ah.

COMMAND-SPECIFIC INFORMATION field

The COMMAND-SPECIFIC INFORMATION field contains information that depends on the command on which the exception condition occurred. When a four byte quantity is stored in the COMMAND-SPECIFIC INFORMATION field, the first four bytes shall be zero.

Further meaning for the COMMAND-SPECIFIC INFORMATION field is defined within the command description in the appropriate command standard (e.g., see SBC-3 for the REASSIGN BLOCKS commands, or SPC-4 the EXTENDED COPY command).

2.4.1.1.4 Sense key specific sense data descriptor

2.4.1.1.4.1 Sense key specific sense data descriptor introduction

The sense key specific sense data descriptor (see table 18) provides additional information about the exception condition. The format and content of the sense-key specific data depends on the value in the SENSE KEY field (see 2.4.1.1.1).

Table 18. Sense key specific sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (02h)							
1	ADDITIONAL LENGTH (06h)							
2	Reserved							
3	Reserved							
4	SKSV (1b)	SENSE KEY SPECIFIC						
5								
6								
7	Reserved							

DESCRIPTOR TYPE and ADDITIONAL LENGTH fields

The DESCRIPTOR TYPE and ADDITIONAL LENGTH fields are described in 2.4.1.1.1. For the sense-key specific sense data descriptor, the DESCRIPTOR TYPE field shall be set to 01h and the ADDITIONAL LENGTH field shall be set to 06h.

SKSV (Sense-key specific valid) bit

The sense-key specific valid (SKSV) bit shall be set to one.

Note. In the fixed format sense data, the SKSV bit indicates whether the contents of the SENSE KEY SPECIFIC field are valid as defined by a command standard. Since the contents of the SENSE KEY SPECIFIC field are valid whenever a sense key specific sense data descriptor is included in the sense data, the only legal value for the SKSV bit is set to one.

The definition of the SENSE KEY SPECIFIC field (see table 19) is determined by the value of the SENSE KEY field (see 2.4.1.1.1).

Table 19. Sense key specific field definitions

Sense Key	Sense Key Specific Field Definition	Reference
ILLEGAL REQUEST	Field pointer	2.4.1.1.4.2
HARDWARE ERROR, MEDIUM ERROR, or RECOVERED ERROR	Actual retry count	2.4.1.1.4.3
NO SENSE or NOT READY	Progress indication	2.4.1.1.4.4
COPY ABORTED	Segment pointer	2.4.1.1.4.5
UNIT ATTENTION	Unit attention condition queue overflow	2.4.1.1.4.6
All other sense keys	The sense key specific sense data descriptor shall not appear in the descriptor format sense data and the SKSV bit (see 2.4.1.2) shall be set to zero in the fixed format sense data.	

2.4.1.1.4.2 Field pointer sense key specific data

If the SENSE KEY is ILLEGAL REQUEST, then the SENSE KEY SPECIFIC field shall be as shown in table 20.

Table 20. Field pointer sense key specific data

Bit / Byte	7	6	5	4	3	2	1	0
0	SKSV (1b)	C/D	Reserved		BPV	BIT POINTER		
1	(MSB)	FIELD POINTER						
2								(LSB)

SKSV bit

The SKSV bit is described in 2.4.1.1.4.1 for descriptor format sense data and in 2.4.1.2 for fixed format sense data.

C/D (Command Data) bit

0 A C/D bit set to zero indicates that the illegal parameter is in the data parameters sent by the application client in the Data-Out Buffer.

1 A C/D bit set to one indicates that the illegal parameter is in the CDB.

BPV (Bit Pointer Valid) bit

0 A BPV bit set to zero indicates that the value in the BIT POINTER field is not valid.

1 A BPV bit set to one indicates that the BIT POINTER field specifies which bit of the byte designated by the FIELD POINTER field is in error. When a multiple-bit field is in error, the BIT POINTER field shall point to the first bit (i.e., the left-most bit) of the field.

FIELD POINTER field

The FIELD POINTER field indicates which byte of the CDB or of the parameter data was in error. Bytes are numbered starting from zero, as shown in the tables describing the commands and parameters. When a multiple-byte field is in error, the field pointer shall point to the first byte (i.e., the left-most byte) of the field. If several consecutive bytes are reserved, each shall be treated as a single-byte field.

Note. The bytes identified as being in error are not necessarily the bytes that need to be changed to correct the problem.

2.4.1.1.4.3 Actual retry count sense key specific data

If the sense key is HARDWARE ERROR, MEDIUM ERROR, or RECOVERED ERROR, then the SENSE KEY SPECIFIC field shall be as shown in table 21.

Table 21. Actual retry count sense key specific data

Bit Byte	7	6	5	4	3	2	1	0
0	SKSV (1b)	Reserved						
1	(MSB)	ACTUAL RETRY COUNT						
2								(LSB)

SKSV bit

The SKSV bit is described in 2.4.1.1.4.1 for descriptor format sense data and in 2.4.1.2 for fixed format sense data.

ACTUAL RETRY COUNT field

The ACTUAL RETRY COUNT field returns vendor specific information on the number of retries of the recovery algorithm used in attempting to recover an error or exception condition.

Note. This field should be computed in the same way as the retry count fields within the Read-Write Error Recovery mode page.

2.4.1.1.4.4 Progress indication sense key specific data

If the sense key is NO SENSE or NOT READY, the SENSE KEY SPECIFIC field shall be as shown in table 22.

Table 22. Progress indication sense key specific data

Bit Byte	7	6	5	4	3	2	1	0
0	SKSV (1b)	Reserved						
1	(MSB)	PROGRESS INDICATION						
2								(LSB)

SKSV bit

The SKSV bit is described in 2.4.1.1.4.1 for descriptor format sense data and in 2.4.1.2 for fixed format sense data.

PROGRESS INDICATION field

The PROGRESS INDICATION field is a percent complete indication in which the returned value is a numerator that has 65 536 (10000h) as its denominator. The progress indication shall be based upon the total operation.

Note. The progress indication should be time related, however this is not an absolute requirement. (E.g., since format time varies with the number of defects encountered, etc., it is reasonable for the device server to assign values to various steps within the process. The granularity of these steps should be small enough to provide reasonable assurances to the application client that progress is being made.)

2.4.1.1.4.5 Segment pointer sense key specific data

If the sense key is COPY ABORTED, the SENSE KEY SPECIFIC field shall be as shown in table 23.

Table 23. Segment pointer sense key specific data

Bit / Byte	7	6	5	4	3	2	1	0
0	SKSV (1b)	Reserved	SD	Reserved	BPV	BIT POINTER		
1	(MSB)	FIELD POINTER						
2								(LSB)

SKSV bit

The SKSV bit is described in 2.4.1.1.4.1 for descriptor format sense data and in 2.4.1.2 for fixed format sense data.

SD (SEGMENT DESCRIPTOR) bit

The SD bit indicates whether the field pointer is relative to the start of the parameter list or to the start of a segment descriptor.

0 An SD bit set to zero indicates that the field pointer is relative to the start of the parameter list.

1 An SD bit set to one indicates that the field pointer is relative to the start of the segment descriptor indicated by the third and fourth bytes of the COMMAND-SPECIFIC INFORMATION field.

BPV (BIT POINTER VALID) bit

0 A BPV bit set to zero indicates that the value in the BIT POINTER field is not valid.

1 A BPV bit set to one indicates that the BIT POINTER field specifies which bit of the byte designated by the FIELD POINTER field is in error.

When a multiple-bit field is in error, the BIT POINTER field shall point to the most-significant (i.e., left-most) bit of the field.

FIELD POINTER field

The FIELD POINTER field indicates which byte of the parameter list or segment descriptor was in error.

If the parameter list is in excess of 65 528 bytes in length and SD is set to zero, the FIELD POINTER value may not fit in two bytes provided by the sense key specific sense data descriptor.

2.4.1.1.4.6 Unit attention condition queue overflow sense key specific data

If the sense key is UNIT ATTENTION, the SENSE KEY SPECIFIC field shall be as shown in table 24

Table 24. Unit attention condition queue overflow sense key specific data

Bit Byte	7	6	5	4	3	2	1	0
0	SKSV (1b)	Reserved						OVERFLOW
1	Reserved							
2								

SKSV bit

The SKSV bit is described in 2.4.1.1.4.1 for descriptor format sense data and in 2.4.1.2 for fixed format sense data.

OVERFLOW bit

0 An OVERFLOW bit set to zero indicates that the unit attention condition queue has not overflowed.

1 An OVERFLOW bit set to one indicates that the unit attention condition queue has overflowed.

2.4.1.1.5 Field replaceable unit sense data descriptor

The field replaceable unit sense data descriptor (see table 25) provides information about a component that has failed.

Table 25. Field replaceable unit sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (03h)							
1	ADDITIONAL LENGTH (02h)							
2	Reserved							
3	FIELD REPLACEABLE UNIT CODE							

DESCRIPTOR TYPE and ADDITIONAL LENGTH fields

The DESCRIPTOR TYPE and ADDITIONAL LENGTH fields are described in 2.4.1.1.1. For the field replaceable unit sense data descriptor, the DESCRIPTOR TYPE field shall be set to 03h and the ADDITIONAL LENGTH field shall be set to 02h.

FIELD REPLACEABLE UNIT CODE field

Non-zero values in the FIELD REPLACEABLE UNIT CODE field are used to identify a component that has failed. A value of zero in this field indicates that no specific component has been identified to have failed or that the data is not available. The format of this information is not specified by this manual. Additional information about the field replaceable unit may be available in the ASCII Information VPD page, if supported by the device server.

2.4.1.1.6 Progress indication sense data descriptor

If the sense key is set to NO SENSE or NOT READY, the progress indication sense data descriptor (see table 26) may be used to provide a progress indication for one operation other than the one described by the non-descriptor fields in 2.4.1.1.1. The sense data should include one progress indication sense data descriptor for each operation for which the device server is able to report progress other than the operation described by the non-descriptor fields in 2.4.1.1.1.

Table 26. Progress indications sense data descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (0Ah)							
1	ADDITIONAL LENGTH (06h)							
2	SENSE KEY							
3	ADDITIONAL SENSE CODE							
4	ADDITIONAL SENSE CODE QUALIFIER							
5	Reserved							
6	MSB	PROGRESS INDICATION						
7								LSB

DESCRIPTOR TYPE and ADDITIONAL LENGTH fields

The DESCRIPTOR TYPE field and ADDITIONAL LENGTH field are described in 2.4.1.1.1.

For the progress indications sense data descriptor:

a) the DESCRIPTOR TYPE field shall be set to 0Ah; and
b) the ADDITIONAL LENGTH field shall be set to 06h.

SENSE KEY field

The SENSE KEY field indicates the operation for which this progress indication sense data descriptor provides a progress indication. A list of sense key values is in 2.4.1.5.

ADDITIONAL SENSE CODE field

The ADDITIONAL SENSE CODE (ASC) field indicates the operation for which this progress indication sense data descriptor provides a progress indication. A list of additional sense codes is in 2.4.1.6.

ADDITIONAL SENSE CODE QUALIFIER field

The ADDITIONAL SENSE CODE QUALIFIER (ASCQ) field indicates detailed information related to the additional sense code for the operation for which this progress indication sense data descriptor provides a progress indication. The value returned in the ADDITIONAL SENSE CODE QUALIFIER (ASCQ) field shall be as specified in 2.4.1.6.

PROGESS INDICATION field

The PROGESS INDICATION field indicates a percent complete for the operation indicated by the SENSE KEY field, the ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field. The value in the PROGESS INDICATION field shall be as defined in 2.4.1.1.4.4.

2.4.1.1.7 Vendor specific sense data descriptors

Vendor specific sense data descriptors (see table 27) contain vendor specific data that further defines the nature of the exception condition.

Table 27. Vendor specific sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (80h - FFh)							
1	ADDITIONAL LENGTH (n-1)							
2	Vendor specific							
n								

DESCRIPTOR TYPE and ADDITIONAL LENGTH fields

The DESCRIPTOR TYPE and ADDITIONAL LENGTH fields are described in 2.4.1.1.1. For the vendor specific sense data descriptor, the DESCRIPTOR TYPE field shall be set to a value between 80h and FFh, inclusive.

2.4.1.2 Fixed format sense data

The fixed format sense data for response codes 70h (current errors) and 71h (deferred errors) is defined in table 28.

Table 28. Fixed format sense data

Bit / Byte	7	6	5	4	3	2	1	0
0	VALID	RESPONSE CODE (70H OR 71H)						
1	Obsolete							
2	FILEMARK	EOM	ILI	RESERVED	SENSE KEY			
3	INFORMATION							
4								
5								
6								
7	ADDITIONAL SENSE LENGTH (N-7)							
8	COMMAND-SPECIFIC INFORMATION							
9								
10								
11								
12	ADDITIONAL SENSE CODE							
13	ADDITIONAL SENSE CODE QUALIFIER							
14	FIELD REPLACEABLE UNIT CODE							
15	SKSV	SENSE KEY SPECIFIC						
16								
17								
18	ADDITIONAL SENSE BYTES							
n								

VALID bit

0 A VALID bit set to zero indicates that the INFORMATION field is not defined in this manual or any command standard.

1 A VALID bit set to one indicates the INFORMATION field contains valid information as defined in this manual or a command standard.

RESPONSE CODE field

The contents of the RESPONSE CODE field indicate the error type and format of the sense data (see 2.4.1). For fixed format sense data, the RESPONSE CODE field shall be set to 70h or 71h.

FILE MARK bit

See the SSC-3 READ and SPACE commands for examples of FILEMARK bit usage.

EOM (End-of-Medium) bit

See the SSC-3 READ, SPACE, and WRITE commands for examples of end-of-medium (EOM) bit usage.

ILI (Incorrect length indicator) Bit

See the SBC-3 READ LONG, SBC-3 WRITE LONG, and SSC-3 READ commands and for examples of incorrect length indicator (ILI) bit usage.

SENSE KEY, ADDITONAL SENSE CODE, and ADDITIONAL SENSE CODE QUALIFIER fields

The SENSE KEY, ADDITIONAL SENSE CODE, and ADDITIONAL SENSE CODE QUALIFIER fields are described in 4.5.2.1.

INFORMATION field

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

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

ADDITIONAL SENSE LENGTH field

The ADDITIONAL SENSE LENGTH field indicates the number of additional sense bytes that follow. The additional sense length shall be less than or equal to 244 (i.e., limiting the total length of the sense data to 252 bytes). If the sense data is being returned as parameter data by a REQUEST SENSE command, then the relationship between the ADDITIONAL SENSE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

COMMAND-SPECIFIC INFORMATION field

The COMMAND-SPECIFIC INFORMATION field contains information that depends on the command on which the exception condition occurred.

FIELD REPLACEABLE UNIT CODE field

The FIELD REPLACEABLE UNIT CODE field is described in 2.4.1.1.5.

SKSV (Sense-key Specific Valid) bit

1 A sense-key specific valid (SKSV) bit set to one indicates the SENSE KEY SPECIFIC field contains valid information as defined in this manual.

0 An SKSV bit set to zero indicates that the SENSE KEY SPECIFIC field is not as defined by this manual.

SENSE KEY SPECIFIC field

The SENSE KEY SPECIFIC field is described in 2.4.1.1.4.

The additional sense bytes may contain vendor specific data that further defines the nature of the exception condition.

2.4.1.3 Current errors

Response codes 70h and 72h (current error) indicate that the sense data returned is the result of an error or exception condition on the task that returned the CHECK CONDITION status or a protocol specific failure condition. This includes errors generated during processing of the command. It also includes errors not related to any command that are detected during processing of a command (e.g., disk servo-mechanism failure, off-track errors, or power-up test errors).

2.4.1.4 Deferred errors

Response codes 71h and 73h (deferred error) indicate that the sense data returned is the result of an error or exception condition that occurred during processing of a previous command for which GOOD, CONDITION MET, INTERMEDIATE, and INTERMEDIATE-CONDITION MET status has already been returned. Such commands are associated with the use of the immediate bit and with some forms of caching. Device servers that implement these features shall implement deferred error reporting.

The deferred error may be indicated by returning CHECK CONDITION status to an application client accessed through a defined I_T nexus as described in this subclause.

If the task terminates with CHECK CONDITION status and the sense data describes a deferred error, the command for the terminated task shall not have been processed. After the device server detects a deferred error condition, it shall return a deferred error according to the following rules:

(a) If no external intervention is necessary to recover a deferred error, a deferred error indication shall not be returned unless required by the error handling parameters of a MODE SELECT command. The occurrence of the error may be logged;

(b) If it is possible to associate a deferred error with an I_T nexus and with a particular function or a particular subset of data, and the error is either unrecovered or required to be reported by the mode parameters, then a deferred error indication shall be returned for a command received on the I_T nexus associated with the deferred error. If an application client request received on an I_T nexus other than the I_T nexus associated with the deferred error attempts to access the particular function or subset of data associated with the deferred error and the TST field equals 000b, then the device server shall respond to the command with a BUSY or ACA ACTIVE status according to the requirements in SAM-4. If an application client request received on an I_T nexus other than the I_T nexus associated with the deferred error attempts to access the particular function or subset of data associated with the deferred error and the TST field equals 001b, then the command attempting the access shall not be blocked by the deferred error and the cause of the deferred error may result in an error being reported for the command attempting the access;

(c) If the device server is unable to associate a deferred error with an I_T nexus or with a particular subset of data, the device server shall return a deferred error for one command received on each I_T nexus. If multiple deferred errors have accumulated for an I_T nexus, only the last error shall be returned;

(d) If the SCSI target device is unable to associate a deferred error with a particular logical unit, it shall establish a deferred error for every logical unit and shall return the deferred error for one command for each logical unit received on each appropriate I_T nexus; or

(e) If a task has never entered the enabled task state, and a deferred error occurs, the task shall be terminated with CHECK CONDITION status and deferred error information returned in the sense data. If a deferred error occurs after a task has entered the enabled task state and the task is affected by the error, the task shall be terminated with CHECK CONDITION status and the current error information shall be returned in the sense data. In this case, if the current error information does not adequately define the deferred error condition, a deferred error may be returned after the current error information has been returned. If a deferred error occurs after a task has entered the enabled task state and the task completes successfully, the device server may choose to return the deferred error information after the completion of the current command in conjunction with a subsequent command that has not begun processing.

Note. A deferred error may indicate that an operation was unsuccessful long after GOOD status was returned. If the application client is unable to replicate or recover from other sources the data that is being written using cached or buffered write operations, then synchronization commands should be performed before the critical data is destroyed. This is necessary for actions taken when deferred errors occur in the storing of the data. The synchronizing process should provide the necessary commands to allow returning CHECK CONDITION status and subsequent returning of deferred error sense information after all cached or buffered operations are completed.

2.4.1.5 Sense key and sense code definitions

The sense keys are defined in table 29.

Table 29. Sense key descriptions (Sheet 1 of 2)

Sense Key	Description
0h	**NO SENSE: Indicates that there is no specific sense key information to be reported. This may occur for a successful command or for a command that receives CHECK CONDITION status because one of the FILEMARK, EOM, or ILI bits is set to one.**
1h	**RECOVERED ERROR:** Indicates that the command completed successfully, with some recovery action performed by the device server. Details may be determined by examining the additional sense bytes and the INFORMATION field. When multiple recovered errors occur during one command, the choice of which error to report (e.g., first, last, most severe) is vendor specific.
2h	**NOT READY:** Indicates that the logical unit is not accessible. Operator intervention may be required to correct this condition.
3h	**MEDIUM ERROR:** Indicates that the command terminated with a non-recovered error condition that may have been caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the device server is unable to distinguish between a flaw in the medium and a specific hardware failure (i.e., sense key 4h).
4h	**HARDWARE ERROR:** Indicates that the device server detected a non-recoverable hardware failure (e.g., controller failure, device failure, or parity error) while performing the command or during a self test.
5h	**ILLEGAL REQUEST:** Indicates that: a) The command was addressed to an incorrect logical unit number (see SAM-4); a) The command had an invalid task attribute (see SAM-4); b) The command was addressed to a logical unit whose current configuration prohibits processing the command; c) There was an illegal parameter in the CDB; or d) There was an illegal parameter in the additional parameters supplied as data for some commands (e.g., PERSISTENT RESERVE OUT). If the device server detects an invalid parameter in the CDB, it shall terminate the command without altering the medium. If the device server detects an invalid parameter in the additional parameters supplied as data, the device server may have already altered the medium.
6h	**UNIT ATTENTION:** Indicates that a unit attention condition has been established (e.g., the removable medium may have been changed, a logical unit reset occurred). See SAM-4.
7h	**DATA PROTECT:** Indicates that a command that reads or writes the medium was attempted on a block that is protected. The read or write operation is not performed.

Table 29. Sense key descriptions (Sheet 2 of 2)

Sense Key	Description
8h	**BLANK CHECK:** Indicates that a write-once device or a sequential-access device encountered blank medium or format-defined end-of-data indication while reading or that a write-once device encountered a non-blank medium while writing.
9h	**VENDOR SPECIFIC:** This sense key is available for reporting vendor specific conditions.
Ah	**COPY ABORTED:** Indicates an EXTENDED COPY command was aborted due to an error condition on the source device, the destination device, or both.
Bh	**ABORTED COMMAND:** Indicates that the device server aborted the command. The application client may be able to recover by trying the command again.
Ch	Obsolete
Dh	**VOLUME OVERFLOW:** Indicates that a buffered SCSI device has reached the end-of-partition and data may remain in the buffer that has not been written to the medium. One or more RECOVER BUFFERED DATA command(s) may be issued to read the unwritten data from the buffer. (See SSC-3.)
Eh	**MISCOMPARE:** Indicates that the source data did not match the data read from the medium.
Fh	Reserved

2.4.1.6 Additional Sense and Additional Sense Qualifier codes

Table 30 lists the Additional Sense (ASC) and Additional Sense Qualifier (ASCQ) codes. Code values are in hexadecimal. ANSI standard SPC-4 lists a more complete table in error description alphabetical order. This Table 30 list adequately covers all Seagate drives, however.

Note. Table 30 is for reference only, as not all drives covered by this manual support all of the codes listed. Codes without sense key references may or may not be supported.

Table 30. Additional Sense and Additional Sense Qualifier codes

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
00	00	No Additional Sense Information	0
01	00	No Index/Logical Block Signal	4
02	00	No SEEK Complete	4
03	00	Peripheral Device Write Fault	1, 3, 4
03	86	Write Fault Data Corruption	
04	00	Logical Unit Not Ready, Cause Not Reportable	2
04	01	Logical Unit Not Ready, Becoming Ready	2
04	02	Logical Unit Not Ready, START UNIT Required	2
04	03	Logical Unit Not Ready, Manual Intervention Required	2
04	04	Logical Unit Not Ready, Format in Progress	2

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
04	**09**	Logical Unit Not Ready, Self Test in Progress	2
04	**0A**	Logical Unit Not Ready, NVC recovery in progress after and exception event	2
04	**11**	Logical Unit Not Ready, Notify (Enable Spinup) required	2
04	**F0**	Logical unit not ready, super certify in progress	2
08	**00**	Logical Unit Communication Failure	9, B
08	**01**	Logical Unit Communication Time-Out	B
08	**02**	Logical Unit Communication Parity Error	
09	**00**	Track Following Error	1, 3, 4
09	**01**	Servo Fault	1, 4
09	**04**	Head Select Fault	3, 4
09	**0D**	Write to at least one copy of a redundant file failed	1
09	**0E**	Redundant files have < 50% good copies	1
09	**F8**	Calibration is needed but the QST is set without the Recal Only bit	1
09	**FF**	Servo Cal completed as part of self-test	1
0A	**00**	Error Log Overflow	
0A	**01**	Failed to write super certify log file	3
0A	**02**	Failed to read super certify log file	3
0B	**00**	Aborted Command	B
0B	**01**	Warning—Specified Temperature Exceeded	1, 6
0B	**02**	Warning, Enclosure Degraded	1
0C	**00**	Write Error	3
0C	**01**	Write Error Recovered With Auto-Reallocation	1
0C	**02**	Write Error—Auto Reallocation Failed	3
0C	**03**	Write Error—Recommend Reassignment	3
0C	**FF**	Write Error—Too many error recovery revs	3
0D	**00**	Volume Overflow Constants	D

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
0E	**00**	Data Miscompare	E
10	**00**	ID CRC Or ECC Error	
11	**00**	Unrecovered Read Error	1, 3
11	**01**	Read Retries Exhausted	
11	**02**	Error Too Long To Correct	
11	**04**	Unrecovered Read Error—Auto Reallocation Failed	3
11	**FF**	Unrecovered Read Error—Too many error recovery revs	3
12	**00**	Address Mark Not Found For ID Field	
12	**01**	Recovered Data Without ECC Using Previous Logical Block ID	
12	**02**	Recovered Data With ECC Using Previous Logical Block ID	
14	**00**	Logical Block Not Found	
14	**01**	Record Not Found	3
15	**00**	Random Positioning Error	
15	**01**	Mechanical Positioning Error	1, 3, 4
15	**02**	Positioning Error Detected By Read Of Medium	
16	**00**	Data Synchronization Mark Error	1, 3, 4
17	**00**	Recovered Data With No Error Correction Applied	
17	**01**	Recovered Data Using Retries	1
17	**02**	Recovered Data Using Positive Offset	1
17	**03**	Recovered Data Using Negative Offset	1
17	**05**	Recovered Data Using Previous Logical Block ID	
17	**06**	Recovered Data Without ECC—Data Auto Reallocated	
18	**00**	Recovered Data With ECC	1
18	**01**	Recovered Data With ECC And Retries Applied	1
18	**02**	Recovered Data With ECC And/Or Retries, Data Auto-Reallocated	1
18	**05**	Recovered Data—Recommand Reassignment	

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
18	06	Recovered Data Using ECC and Offsets	
18	07	Recovered Data With ECC—Data Rewritten	1
19	00	Defect List Error	1, 4
19	01	Defect List Not Available	
19	02	Defect List Error In Primary List	
19	03	Defect List Error in Grown List	
19	0E	Fewer than 50% Defect List Copies	
1A	00	Parameter List Length Error	5
1B	00	Synchronous Data Transfer Error	
1C	00	Defect List Not Found	1, 4
1C	01	Primary Defect List Not Found	
1C	02	Grown Defect List Not Found	
1C	83	Seagate Unique Diagnostic Code	
1D	00	Miscompare During Verify Operation	E
1F	00	Number of Defects Overflows the Allocated Space That The Read Defect Command Can Handle	1
20	00	Invalid Command Operation Code	5
20	F3	Invalid linked command operation code	5
21	00	Logical Block Address Out Of Range	D
24	00	Invalid Field In CDB	5
24	01	Illegal Queue Type for CDB (Low priority commands must be SIMPLE queue)	5
24	F0	Invalid LBA in linked command	5
24	F2	Invalid linked command operation code	5
24	F3	Illegal G->P operation request	5
25	00	Logical Unit Not Supported	5
26	00	Invalid Field In Parameter List	5
26	01	Parameter Not Supported	5

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
26	**02**	Parameter Value Invalid	5
26	**03**	Invalid Field Parameter—Threshold Parameter	5
26	**04**	Invalid Release of Active Persistent Reserve	5
26	**05**	Fail to read valid log dump data	5
26	**97**	Invalid Field Parameter—TMS Firmware Tag	
26	**98**	Invalid Field Parameter—Check Sum	
26	**99**	Invalid Field Parameter—Firmware Tag	
27	**00**	Write Protected	7
29	**00**	Flashing LED occurred	4
29	**00**	Power On, Reset, Or Bus Device Reset Occurred	6
29	**01**	Power-On Reset Occurred	6
29	**02**	SCSI Bus Reset Occurred	6
29	**03**	Bus Device Reset Function Occurred	6
29	**04**	Internal Reset Occurred	6
29	**05**	Transceiver Mode Changed To Single-Ended	6
29	**06**	Transceiver Mode Changed To LVD	6
29	**07**	Write Log Dump data to disk successful OR IT Nexus Loss	6
29	**08**	Write Log Dump data to disk fail	6
29	**09**	Write Log Dump Entry information fail	6
29	**0A**	Reserved disc space is full	6
29	**0B**	SDBP test service contained an error, examine status packet(s) for details	6
29	**0C**	SDBP incoming buffer overflow (incoming packet too big)	6
29	**CD**	Flashing LED occurred. (Cold reset)	6
29	**CE**	Flashing LED occurred. (Warm reset)	6
2A	**01**	Mode Parameters Changed	6
2A	**02**	Log Parameters Changed	6

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
2A	**03**	Reservations preempted	6
2A	**04**	Reservations Released	6
2A	**05**	Registrations Preempted	6
2C	**00**	Command Sequence Error	5
2F	**00**	Tagged Commands Cleared By Another Initiator	6
31	**00**	Medium Format Corrupted	3
31	**01**	Corruption in R/W format request	3
31	**91**	Corrupt World Wide Name (WWN) in drive information file	3
32	**00**	No Defect Spare Location Available	4
32	**01**	Defect List Update Error	3, 4, 5
32	**02**	No Spares Available—Too Many Defects On One Track	
32	**03**	Defect list longer than allocated memory	3
33	**00**	Flash not ready for access	3
35	**00**	Unspecified Enclosure Services Failure	4
35	**01**	Unsupported Enclosure Function	5
35	**02**	Enclosure Services Unavailable	2
35	**03**	Enclosure Transfer Failure	4
35	**04**	Enclosure Transfer Refused	4
37	**00**	Parameter Rounded	1
3D	**00**	Invalid Bits In Identify Message	
3E	**03**	Logical Unit Failed Self Test	4
3E	**00**	Logical Unit Has Not Self Configured Yet	
3F	**00**	Target Operating Conditions Have Changed	6
3F	**01**	Device internal reset occurred	6
3F	**02**	Changed Operating Definition	6
3F	**05**	Device Identifier Changed	6

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
3F	**0F**	Echo buffer overwritten	B
3F	**80**	Buffer contents have changed	1
3F	**90**	Invalid APM Parameters	
3F	**91**	World Wide Name (WWN) Mismatch	6
40	**01**	DRAM Parity Error	1, 4
40	**02**	Spinup Error recovered with retries	1
42	**00**	Power-On Or Self-Test Failure	4
42	**0A**	Port A failed loopback test	4
42	**0B**	Port B failed loopback test	4
43	**00**	Message Reject Error	B
44	**00**	Internal Target Failure	1, 3, 4
44	**F2**	Data Integrity Check Failed on verify	4
44	**F6**	Data Integrity Check Failed during write	4
44	**FF**	XOR CDB check error	4
45	**00**	Select/Reselection Failure	B
47	**00**	SCSI Parity Error	B
47	**03**	Information Unit CRC Error	B
47	**80**	Fibre Channel Sequence Error	B
48	**00**	Initiator Detected Error Message Received	B
49	**00**	Invalid Message Received	B
4B	**00**	Data Phase Error	B
4B	**01**	Invalid transfer tag	B
4B	**02**	Too many write data	B
4B	**03**	ACK NAK Timeout	B
4B	**04**	NAK received	B
4B	**05**	Data Offset error	B

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
4B	**06**	Initiator response timeout	B
4C	**00**	Logical Unit Failed Self-Configuration	
4E	**00**	Overlapped Commands Attempted	B
55	**01**	XOR Cache is Not Available	
55	**04**	PRKT table is full	5
5B[1]	**00**	Log Exception	
5B*	**01**	Threshold Condition Met	
5B*	**02**	Log Counter At Maximum	
5B*	**03**	Log List Codes Exhausted	
5C	**00**	RPL Status Change	6
5C	**01**	Spindles Synchronized	
5C	**02**	Spindles Not Synchronized	
5D	**00**	Failure Prediction Threshold Exceeded	1, 6
5D	**FF**	False Failure Prediction Threshold Exceeded	1, 6
65	**00**	Voltage Fault	4
80	**00**	General Firmware Error Qualifier	9
80	**86**	IOEDC Error on Read	9
80	**87**	IOEDC Error on Write	9
80	**88**	Host Parity Check Failed	9
80	**89**	IOEDC Error on Read Detected by Formatter	9
80	**8A**	Host FIFO Parity Error detected by Common Buffer	9

ASC (byte 12)	ASCQ (byte 13)	Description	Sense Key
80	**8B**	Host FIFO Parity Error detected by frame buffer logic	9
80	**8C**	Host Data Frame Buffer Parity Error	9
81	**00**	Reassign Power—Fail Recovery Failed	
81	**00**	LA Check Error, LCM bit = 0	4
81	**00**	LA Check Error	B
B4	**00**	Unreported Deferred Errors have been logged on log page 34h	6

[1] Can be supported, but is a factory installed option.

3.0 Command Reference

This clause describes the commands supported by Seagate Disc Drives.

Commands that have been supported prior to this manual being produced and that are now indicated as obsolete will be described in this clause as obsolete. However, a description will be provided for such commands.

Commands that have been declared obsolete by the T10 committee and were never supported by Seagate are not included in this manual. Commands that have not been supported in the past, but may be supported in the near future will be included in this manual.

This clause contains information about the commands used by Seagate Parallel SCSI, Fibre Channel, and Serial Attached SCSI disc drives. This clause is organized to provide rapid access to command information.

Two types of commands are supported by the drive: commands for all devices; and commands for direct access devices. The individual Product Manuals for each Seagate model drive list the Commands and parameter pages that the particular drive supports.

Commands sorted by command name

Command name	OP code [1]	Command type		Length					clause Reference
		All devices	Direct access devices	6 byte	10 byte	12 byte	16 byte	32 byte	
CHANGE DEFINITION	40h	x [2]							3.1
COMPARE	39h	x [2]							3.2
COPY	18h	x [2]							3.3
COPY AND VERIFY	3Ah	x [2]							3.4
FORMAT UNIT	04h		x	x					3.5
INQUIRY	12h	x		x					3.6
LOCK-UNLOCK CACHE (10)	36h		x		x				3.7
LOCK-UNLOCK CACHE (16)	92h		x				x		3.8
LOG SELECT	4Ch	x			x				3.9
LOG SENSE	4Dh	x			x				3.10
MODE SELECT (6)	15h		x	x					3.11
MODE SELECT (10)	55h		x		x				3.12
MODE SENSE (6)	1Ah		x	x					3.13
MODE SENSE (10)	5Ah		x		x				3.14

Command name	OP code [1]	Command type		Length					clause Reference
		All devices	Direct access devices	6 byte	10 byte	12 byte	16 byte	32 byte	
PERSISTENT RESERVE IN	5Eh		x						3.15
PERSISTENT RESERVE OUT	5Fh		x						3.16
READ (6)	08h		x	x					3.19
READ (10) [3]	28h		x		x				3.20
READ (12)	A8h		x			x			3.21
READ (16)	88h		x				x		3.22
READ (32)	7Fh/0009h		x					x	3.23
READ BUFFER	3Ch	x			x				3.24
READ CAPACITY (10)	25h		x		x				3.25
READ CAPACITY (16)	9Eh/10h		x				x		3.26
READ DEFECT DATA (10)	37h		x		x				3.27
READ DEFECT DATA (12)	B7h		x			x			3.28
READ LONG (10)	3Eh		x		x				3.29
READ LONG (16)	9Eh		x				x		3.30
REASSIGN BLOCKS	07h		x	x					3.31
RECEIVE DIAGNOSTIC RESULTS	1Ch	x		x					3.32
RELEASE (6)	17h		x [2]	x					3.33
RELEASE (10)	57h		x [2]		x				3.34
REPORT DEVICE IDENTIFIER	A3h/05h	x				x			3.35
REPORT LUNS	A0h		x						3.37
REQUEST SENSE	03h	x		x					3.41
RESERVE (6)	16h		x [2]	x					3.42
RESERVE (10)	56h		x [2]		x				3.43
REZERO UNIT	01h		x [2]	x					3.44
SEEK (6)	0Bh		x [2]	x					3.47
SEEK EXTENDED (10)	2Bh		x		x				3.48
SEND DIAGNOSTIC	1Dh	x		x					3.49
SET DEVICE IDENTIFIER	A4h/06h	x				x			3.50
START/STOP UNIT	1Bh		x	x					3.52
SYNCHRONIZE CACHE (10)	35h		x		x				3.53
SYNCHRONIZE CACHE (16)	91h		x				x		3.54
TEST UNIT READY	00h	x		x					3.55

Command name	OP code [1]	Command type		Length					clause Reference
		All devices	Direct access devices	6 byte	10 byte	12 byte	16 byte	32 byte	
UNMAP	42h		x		x				3.56
VERIFY(10)	2Fh		x		x				3.57
VERIFY (12)	AFh		x			x			3.58
VERIFY (16)	8Fh		x				x		3.59
VERIFY (32)	7Fh/000Ah		x					x	3.60
WRITE (6)	0Ah		x	x					3.61
WRITE (10)	2Ah		x		x				3.62
WRITE (12)	AAh		x			x			3.63
WRITE (16)	8Ah		x				x		3.64
WRITE (32)	7Fh/000Bh		x					x	3.65
WRITE AND VERIFY (10)	2Eh		x		x				3.66
WRITE AND VERIFY (12)	AEh		x			x			3.67
WRITE AND VERIFY (16)	8Eh		x				x		3.68
WRITE AND VERIFY (32)	7Fh/000Ch		x					x	3.69
WRITE BUFFER	3Bh	x			x				3.70
WRITE LONG	3Fh		x		x				3.71
WRITE LONG (16)	9Fh/11h		x				x		3.72
WRITE SAME (10)	41h		x		x				3.73
WRITE SAME (16)	93h		x				x		3.74
WRITE SAME (32)	7Fh/000Dh		x					x	3.75

[1] Command Operation Code a one byte hexidecimal number between 00h and FF hex.

[2] Command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

[3] Formerly called Read Extended.

3.1 CHANGE DEFINITION command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

The CHANGE DEFINITION command (see table 31) is used to modify the operating definition of the device server(s) with respect to commands from the sending initiator or with respect to commands from all initiators.

Table 31. CHANGE DEFINITION command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (40h)							
3	Reserved							
4	Reserved							SAVE
5	Reserved	DEFINITION PARAMETER						
6	Reserved							
7	Reserved							
8	Reserved							
9	Reserved							
10	PARAMETER DATA LENGTH							
11	CONTROL							

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

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

The definition parameter field is defined in table 14.

Table 32. Definition parameter field

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

The current operating definition parameter values establish operating definitions compatible with the applicable SCSI standard. Definitions supported by an SCSI device are returned in the implemented operating definition page (see 8.4.4 in SCSI-3 Primary Commands Rev 11).

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

The parameter data is vendor-specific.

Note. 1 The parameter data may be used to specify a password to validate an operating definition change.

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

(a) Change the operating definition of a logical unit relative to the initiator that issued the command: In this case, the target is capable of maintaining a separate operating definition for each logical unit relative to each initiator in the system;

(b) Change the operating definition of all logical units in the target relative to the initiator that issued the command: In this case, the target is capable of maintaining a unique operating definition, for each initiator in the system, that applies to all logical units in the target;

(c) Change the operating definition of a logical unit relative to all initiators in the system: In this case, the target is capable of maintaining a separate operating definition for each logical unit relative to all initiators in the system; or

(d) Change the operating definition of all logical units in the target relative to all initiators in the system: In this case, the target is capable of maintaining only one operating definition.

Note. 2 This manual does not provide a direct means to determine which of the above four methods has been implemented. An indirect means of determining which method is implemented exists in that the device server is required to inform affected initiators of operating definition changes via the unit attention condition.

Note. 3 The modifications listed c) and d) above may result in incompatibilities if other initiators are using a different SCSI version.

The operating definition is modified after successful completion of the command. The application client should verify the new operating definition by issuing an INQUIRY command requesting the implemented operating definition page (see 8.4.1 in SCSI -3).

Note. 4 The method of changing the operating definition is vendor-specific. Some implementations may require that the target's operating mode be reinitialized as if a power-up or hard reset had occurred. Other implementations may modify only those operating definitions that are affected by the CHANGE DEFINITION command.

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

Note. 5 The present operating definition of the target may always be interrogated through the INQUIRY command. When an SCSI-3 target has its operating definition changed to an older SCSI operating definition, certain changes are needed to promote compatibility with preexisting older SCSI initiators.

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

3.2 COMPARE command

Obsolete. If received, a CHECK CONDITION status is sent.

3.3 COPY command

Obsolete. If received, a CHECK CONDITION status is sent.

3.4 COPY AND VERIFY command

Obsolete. If received, a CHECK CONDITION status is sent.

3.5 FORMAT UNIT command

3.5.1 FORMAT UNIT command overview

The FORMAT UNIT command (see table 33) requests that the device server format the medium into application client accessible logical blocks as specified in the number of blocks and block length values received in the last mode parameter block descriptor (see 4.3.4) in a MODE SELECT command (see 3.11and 3.12). In addition, the device server may certify the medium and create control structures for the management of the medium and defects. The degree that the medium is altered by this command is vendor-specific.

If a device server receives a FORMAT UNIT command before receiving a MODE SELECT command with a mode parameter block descriptor the device server shall use the number of blocks and block length at which the logical unit is currently formatted (i.e., no change is made to the number of blocks and the block length of the logical unit during the format operation).

If any deferred downloaded code has been received as a result of a WRITE BUFFER command (see 3.70), then that deferred downloaded code shall replace the current operational code.

Table 33. FORMAT UNIT command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (04h)							
1	FMTPINFO		LONGLIST	FMTDATA	CMPLIST	DEFECT LIST FORMAT		
2	Vendor Specific							
3	Obsolete							
4								
5	CONTROL							

The simplest form of the FORMAT UNIT command (i.e., a FORMAT UNIT command with no parameter data) accomplishes medium formatting with little application client control over defect management. The device server implementation determines the degree of defect management that is to be performed. Additional forms of this command increase the application client's control over defect management. The application client may specify:

a) defect list(s) to be used;

b) defect locations;

c) that logical unit certification be enabled; and

d) exception handling in the event that defect lists are not accessible.

While performing a format operation, the device server shall respond to commands attempting to enter into the task set except INQUIRY commands, REPORT LUNS commands, and REQUEST SENSE commands 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. Handling of commands already in the task set is vendor-specific. If the device server receives an INQUIRY command, a REPORT LUNS commands, or a REQUEST SENSE command, then the device server shall process the command. The device server shall return data for an INQUIRY command based on the condition of the SCSI target device before beginning the FORMAT UNIT command (i.e., INQUIRY data shall not change until after successful completion of a format operation). The processing of commands in the task set when a FORMAT UNIT command is received is vendor specific.

PROGRESS INDICATION field

The PROGRESS INDICATION field in parameter data returned in response to a REQUEST SENSE command (see 3.41) may be used by the application client at any time during a format operation to poll the logical unit's progress. While a format operation is in progress unless an error has occurred, a device server shall respond to a REQUEST SENSE command by returning parameter data containing sense data with the sense key set to NOT READY and the additional sense code set to LOGICAL UNIT NOT READY, FORMAT IN PROGRESS with the sense key specific bytes set for progress indication (see 2.4.1.1.6).

FMTPINFO (Format Protection Information) field

The format protection information (FMTPINFO) field (see table 38) in combination with the PROTECTION FIELD USAGE field (see 3.5.3) specifies whether or not the device server enables or disables the use of protection information.

When protection information is written during a FORMAT UNIT command (i.e., the FMTPINFO bit is set to one) protection information shall be written to a default value of FFFFFFFF_FFFFFFFFh.

LONGLIST bit

0 A LONGLIST bit set to zero specifies that the parameter list, if any, contains a short parameter list header as defined in table 36.

1 A LONGLIST bit set to one specifies that the parameter list, if any, contains a long parameter list header as defined in table 37. If the FMTDATA bit is set to zero, the LONGLIST bit shall be ignored.

FMTDATA (Format Data)

0 A format data (FMTDATA) bit set to zero specifies that no parameter list be transferred from the data-out buffer.

1 A FMTDATA bit set to one specifies that the FORMAT UNIT parameter list (see table 35) shall be transferred from the data-out buffer. The parameter list consists of a parameter list header, followed by an optional initialization pattern descriptor, followed by an optional defect list.

CMPLST (Complete List)

0 A complete list (CMPLST) bit set to zero specifies that the defect list included in the FORMAT UNIT parameter list shall be used in an addition to the existing list of defects. As a result, the device server shall construct a new GLIST that contains:

a) the existing GLIST;

b) the DLIST, if it is sent by the application client; and

c) the CLIST, if certification is enabled (i.e., the device server may add any defects it detects during the format operation).

1 A CMPLST bit set to one specifies that the defect list included in the FORMAT UNIT parameter list is a complete list of defects. Any existing defect list except the PLIST shall be ignored by the device server. As a result, the device server shall construct a new GLIST that contains:

a) the DLIST, if it is sent by the application client; and

b) the CLIST, if certification is enabled (i.e., the device server may add any defects it detects during the format operation).

Note. If the FMTDATA bit is set to zero, the CMPLIST bit shall be ignored.

DEFECT LIST FORMAT field

The DEFECT LIST FORMAT field specifies the format of the address descriptors in the defect list if the FMTDATA bit is set to one (see table 34).

Table 34 defines the address descriptor usage for the FORMAT UNIT command.

Table 34. FORMAT UNIT command address descriptor usage

Field in the FORMAT UNIT CDB			DEFECT LIST LENGTH field in the parameter list header	Type [a]	Comments [f]
FMTDATA	CMPLST	DEFECT LIST FORMAT			
0	Any	000b	Not available	M	Vendor-specific defect information
1	0	000b (short block)	Zero	O	See [b] and [d]
	1			O	See [b] and [e]
	0		Nonzero	O	See [c] and [d]
	1			O	See [b] and [e]
1	0	011b (long block)	Zero	O	See [b] and [d]
	1			O	See [b] and [e]
	0		Nonzero	O	See [c] and [d]
	1			O	See [c] and [e]

a M = implementation is mandatory. O = implementation is optional.

b No DLIST is included in the parameter list.

c A DLIST is included in the parameter list. The device server shall add the DLIST defects to the new GLIST.

d The device server shall add existing GLIST defects to the new GLIST (i.e., use the existing GLIST).

e The device server shall not add existing GLIST defects to the new GLIST (i.e., discard the existing GLIST).

f All the options described in this table cause a new GLIST to be created during processing of the FORMAT UNIT command as described in the text.

Table 34. FORMAT UNIT command address descriptor usage

Field in the FORMAT UNIT CDB			DEFECT LIST LENGTH field in the parameter list header	Type [a]	Comments [f]
FMTDATA	CMPLST	DEFECT LIST FORMAT			
1	0	100b (bytes from index)	Zero	O	
	1			O	
	0		Nonzero	O	
	1			O	
1	0	101b (physical sector)	Zero	O	
	1			O	
	0		Nonzero	O	
	1			O	
1	0	110b (vendor specific)	Vendor specific	O	
	1			O	
All others				Reserved.	

[a] M = implementation is mandatory. O = implementation is optional.

[b] No DLIST is included in the parameter list.

[c] A DLIST is included in the parameter list. The device server shall add the DLIST defects to the new GLIST.

[d] The device server shall add existing GLIST defects to the new GLIST (i.e., use the existing GLIST).

[e] The device server shall not add existing GLIST defects to the new GLIST (i.e., discard the existing GLIST).

[f] All the options described in this table cause a new GLIST to be created during processing of the FORMAT UNIT command as described in the text.

3.5.2 FORMAT UNIT parameter list

3.5.2.1 FORMAT UNIT parameter list overview

Table 35 defines the FORMAT UNIT parameter list.

Table 35. FORMAT UNIT parameter list

Bit Byte	7	6	5	4	3	2	1	0
0 to 3 or 0 to 7	Parameter List Header (see table 36 or table 37 in 3.5.3)							
	INitialization Pattern Descriptor (if any) (see table 39, in 3.5.4)							
	Defect List (if any)							

PARAMETER LIST HEADER field

The PARAMETER LIST HEADER is defined in 3.5.3.

INITIALIZATION PATTERN DESCRIPTOR field

The INITIALIZATION PATTERN DESCRIPTOR, if any, is defined in 3.5.4.

DEFECT LIST field

The DEFECT LIST, if any, contains address descriptors (see table 34) each specifying a location on the medium that the device server shall exclude from the application client accessible part. This is called the DLIST.

3.5.3 Parameter list header

The parameter list headers (see table 36 and table 37) provide several optional format control parameters. Device servers that implement these headers provide the application client additional control over the use of the four defect sources, and the format operation. If the application client attempts to select any function not implemented by the device server, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The short parameter list header (see table 36) is used if the LONGLIST bit is set to zero in the FORMAT UNIT CDB.

Table 36. Short parameter list header

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="5">Reserved</td><td colspan="3">PROTECTION FIELDS USAGE</td></tr>
<tr><td>1</td><td>FOV</td><td>DPRY</td><td>DCRT</td><td>STPF</td><td>IP</td><td>Obsolete</td><td>IMMED</td><td>Vendor specific</td></tr>
<tr><td>2</td><td>(MSB)</td><td colspan="6" rowspan="2">DEFECT LIST LENGTH</td><td></td></tr>
<tr><td>3</td><td></td><td>(LSB)</td></tr>
</table>

The long parameter list header (see table 37) is used if the LONGLIST bit is set to one in the FORMAT UNIT CDB.

Table 37. Long parameter list header

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="5">Reserved</td><td colspan="3">PROTECTION FIELDS USAGE</td></tr>
<tr><td>1</td><td>FOV</td><td>DPRY</td><td>DCRT</td><td>STPF</td><td>IP</td><td>Obsolete</td><td>IMMED</td><td>Vendor specific</td></tr>
<tr><td>2</td><td colspan="8">Reserved</td></tr>
<tr><td>3</td><td colspan="4">P_I_INFORMATION</td><td colspan="4">PROTECTION INTERVAL EXPONENT</td></tr>
<tr><td>4</td><td>(MSB)</td><td colspan="6" rowspan="2">DEFECT LIST LENGTH</td><td></td></tr>
<tr><td>7</td><td></td><td>(LSB)</td></tr>
</table>

The PROTECTION FIELD USAGE field in combination with the FMTPINFO field (see table 38) specifies the requested protection type (see SBC-3).

Table 38. FMTPINFO field and PROTECTION FIELDS USAGE field

Device server indication		Application client specification		Description
SPT [a]	**PROTECT** [b]	**FMTPINFO**	**PROTECTION FIELD USAGE**	
xxxb	0	00b	000b	The logical unit shall be formatted to type 0 protection [c] (see SBC-3) resulting in the P_TYPE field [d] being set to 000b.
xxxb	0	00b	>000b	Illegal [e]
xxxb	0	01b	xxxb	Illegal [f]
xxxb	0	1xb	xxxb	Illegal [f]
xxxb	1	00b	000b	The logical unit shall be formatted to type 0 protection [c] (see SBC-3) resulting in the P_TYPE field [d] being set to 000b.
xxxb	1	00b	>000b	Illegal [e]
xxxb	1	0b	xxxb	Illegal [f]
000b 001b 011b 111b	1	10b	000b	The logical unit shall be formatted to type 1 protection [g] (see SBC-3) resulting in the P_TYPE field [d] being set to 000b.
000b 001b 011b 111b	1	10b	>000b	Illegal [e]
000b	1	11b	xxxb	Illegal [f]
000b 001b 011b 111b	1	11b	000b	The logical unit shall be formatted to type 2 protection [g] (see SBC-3) resulting in the P_TYPE field [d] being set to 001b.
001b 010b	1	11b	>000b	Illegal [e]
011b 100b	1	11b	000b	Illegal [e]
011b 100b 101b 111b	1	11b	001b	The logical unit shall be formatted to type 3 protection. [g] (see SBC-3) resulting in the P_TYPE field [d] being set to 010b.

Table 38. FMTPINFO field and PROTECTION FIELDS USAGE field

Device server indication		Application client specification		Description
SPT [a]	PROTECT [b]	FMTPINFO	PROTECTION FIELD USAGE	
011b 100b 101b 111b	1	11b	>001b	Illegal [e]
110b	1	1xb	xxxb	Reserved

a See the Extended INQUIRY Data VPD page(see 4.4.6) for the definition of the spt field.
b See the standard INQUIRY data(see 3.6.2) for the definition of the protect bit.
c The device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header (see 4.3.3).
d See the READ CAPACITY command (see 3.25.1) for the definition of the P_TYPE field.
e The device server 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.
f The device server 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.
g The device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the block length is 512, then the formatted block length is 520). Following a successful format, the prot_en bit in the READ CAPACITY (16) parameter data (see 3.26.1) indicates whether protection information (see SBC-3) is enabled.

FOV (Format Options Valid) bit

0 A format options valid (FOV) bit set to zero specifies that the device server shall use its default settings for the DPRY, DCRT, STPF, and IP bits. If the FOV bit is set to zero, the application client shall set these bits to zero. If the FOV bit is set to zero and any of the other bits listed in this paragraph are not set to zero, the device server 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.

1 A FOV bit set to one specifies that the device server shall examine the values of the DPRY, DCRT, STPF, and IP bits. When the FOV bit is set to one, the DPRY, DCRT, STPF, and IP bits are defined as follows.

DPRY (Disable Primary) bit

0 A disable primary (DPRY) bit set to zero specifies that the device server shall not use parts of the medium identified as defective in the PLIST for application client accessible logical blocks. If the device server is not able to locate the PLIST or it is not able to determine whether a PLIST exists, it shall take the action specified by the STPF bit.

1 A DPRY bit set to one specifies that the device server shall not use the PLIST to identify defective areas of the MEDIUM. The PLIST shall not be deleted.

DCRT (disable certification) bit

0 A disable certification (DCRT) bit set to zero specifies that the device server shall perform a vendor-specific medium certification operation to generate a CLIST.

1 A DCRT bit set to one specifies that the device server shall not perform any vendor-specific medium certification process or format verification operation.

STPF (Stop Format) bit

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

a) The device server has been requested to use the PLIST (i.e., the DPRY bit is set to zero) or the GLIST (i.e., the CMPLST bit is set to zero) and the device server is not able to locate the list or determine whether the list exists; or

b) The device server has been requested to use the PLIST (i.e., the DPRY bit is set to zero) or the GLIST (i.e., the CMPLST bit is set to zero), and the device server encounters an error while accessing the defect list.

0 A STPF bit set to zero specifies that, if one or both of these events occurs, the device server shall continue to process the FORMAT UNIT command. The device server shall return CHECK CONDITION status at the completion of the FORMAT UNIT command with the sense key set to RECOVERED ERROR and the additional sense code set to either DEFECT LIST NOT FOUND if the condition described in item a) occurred, or DEFECT LIST ERROR if the condition described in item b) occurred.

1 A STPF bit set to one specifies that, if one or both of these events occurs, the device server shall terminate the FORMAT UNIT command with CHECK CONDITION status and the sense key shall be set to MEDIUM ERROR with the additional sense code set to either DEFECT LIST NOT FOUND if the condition described in item a) occurred, or DEFECT LIST ERROR if the condition described in item b) occurred.

Note. The use of the FMTDATA bit, the CMPLST bit, and the parameter list header allow the application client to control the source of the defect lists used by the FORMAT UNIT command. Setting the DEFECT LIST LENGTH field to zero allows the application client to control the use of PLIST and CLIST without having to specify a DLIST.

IP (initialization pattern) bit

0 An initialization pattern (IP) bit set to zero specifies that an initialization pattern descriptor is not included and that the device server shall use its default initialization pattern.

1 An IP bit set to one specifies that an initialization pattern descriptor (see 3.5.4) is included in the FORMAT UNIT parameter list following the parameter list header.

IMMED (Immediate) bit

0 An immediate (IMMED) bit set to zero specifies that the device server shall return status after the format operation has completed.

1 An IMMED bit value set to one specifies that the device server shall return status after the entire parameter list has been transferred.

The P_I_INFORMATION field shall be set to zero.

For a type 1 protection information request, if the PROTECTION INTERVAL EXPONENT field is not set to zero, then the device server 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.

For a type 2 protection or a type 3 protection format request, the protection interval exponent determines the length of user data to be sent before protection information is transferred (i.e., the protection information interval).

The protection information interval is calculated as follows:

$$\text{protection information interval} = \text{logical block length} / 2^{(\text{protection interval exponent})}$$

where:

logical block length	is the length in bytes of a logical block as specified in the mode parameter block descriptor (see clause 4.3.4.1)
protection interval exponent	is the contents of the PROTECTION INTERVAL EXPONENT field

If the protection information interval calculates to a value that is not an even number (e.g., $520/2^3 = 65$) or not a whole number (e.g., $520/2^4 = 32.5$ and $520/2^{10} = 0.508$), then the device server 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.

DEFECT LIST LENGTH field

The DEFECT LIST LENGTH field specifies the total length in bytes of the defect list (i.e., the address descriptors) that follows and does not include the initialization pattern descriptor, if any. The formats for the address descriptor(s) are shown in table 42.

Short block format address descriptors and long block format address descriptors should be in ascending order. Bytes from index format address descriptors and physical sector format address descriptors shall be in ascending order. More than one physical or logical block may be affected by each address descriptor. If the address descriptors are not in the required order, the device server 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.

3.5.4 Initialization pattern descriptor

The initialization pattern descriptor specifies that the device server initialize logical blocks to a specified pattern. The initialization pattern descriptor (see table 39) is sent to the device server as part of the FORMAT UNIT parameter list.

Table 39. Initialization pattern descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	IP MODIFIER		SI	Reserved				
1	INITIALIZATION PATTERN TYPE							
2	(MSB)	INITIALIZATION PATTERN LENGTH (N - 3)						
3								(LSB)
4	INITIALIZATION PATTERN							
N								

IP MODIFIER (Initilization Pattern Modifier) field

The initialization pattern modifier (IP MODIFIER) field (see table 40) specifies the type and location of a header that modifies the initialization pattern.

Table 40. Initialization pattern modifier (IP MODIFIER) field

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

SI (Security Initialize) bit

A security initialize (SI) bit set to one specifies that the device server shall attempt to write the initialization pattern to all areas of the medium including those that may have been reassigned (i.e., are in a defect list). An SI bit set to one shall take precedence over any other FORMAT UNIT CDB field. The initialization pattern shall be written using a security erasure write technique. Application clients may choose to use this command multiple times to fully erase the previous data. Such security erasure write technique procedures are outside the scope of this manual. The exact requirements placed on the security erasure write technique are vendor-specific. The intent of the security erasure write is to render any previous user data unrecoverable by any analog or digital technique.

0 An SI bit set to zero specifies that the device server shall initialize the application client accessible part of the medium. The device server is not required to initialize other areas of the medium. However, the device server shall format the medium as defined in the FORMAT UNIT command.

1 When the SI bit is set to one, the device server need not write the initialization pattern over the header and other header and other parts of the medium not previously accessible to the application client. If the device server is unable to write over any part of the medium that is currently accessible to the application client or may be made accessible to the application client in the future (e.g., by clearing the defect list), it shall terminate the command with CHECK CONDITION status with the sense key set to MEDIUM ERROR and the additional sense code set to the appropriate value for the condition. The device server shall attempt to rewrite all remaining parts of the medium even if some parts are not able to be rewritten.

INITIALIZATION PATTERN TYPE field

The INITIALIZATION PATTERN TYPE field (see table 41) specifies the type of pattern the device server shall use to initialize each logical block within the application client accessible part of the medium. All bytes within a logical block shall be written with the initialization pattern. The initialization pattern is modified by the IP MODIFIER field as described in table 40.

Table 41. INITIALIZATION PATTERN TYPE field

Code	Description
00h	Use a default initialization pattern [a]
01h	Repeat the pattern specified in the INITIALIZATION PATTERN field as required to fill the logical block [b]
02h - 7Fh	Reserved
80h - FFh	Vendor-specific

a If the INITIALIZATION PATTERN LENGTH field is not set to zero, the device server 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.

b If the INITIALIZATION PATTERN LENGTH field is set to zero, the device server 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.

INITIALIZATION PATTERN LENGTH field

The INITIALIZATION PATTERN LENGTH field specifies the number of bytes contained in the INITIALIZATION PATTERN field. If the initialization pattern length exceeds the current block length the device server 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.

INITIALIZATION PATTERN field

The INITIALIZATION PATTERN field specifies the initialization pattern. The initialization pattern is modified by the IP MODIFIER field.

3.5.5 Address descriptor formats

3.5.5.1 Address descriptor formats overview

This subclause describes the address descriptor formats used in the FORMAT UNIT command, the READ DEFECT DATA commands (see 3.27 and 3.28), and the Translate Address diagnostic pages (see 4.1.3) of the SEND DIAGNOSTIC command and the RECEIVE DIAGNOSTIC RESULTS command.

The format type of an address descriptor is specified with:

a) the DEFECT LIST FORMAT field in the CDB, for the FORMAT UNIT command and the READ DEFECT DATA commands;

b) the SUPPLIED FORMAT field, for the Translate Address diagnostic pages; or

c) the TRANSLATE FORMAT field, for the Translate Address diagnostic pages.

Table 42 defines the types of address descriptors.

Table 42. Address descriptor formats

Format type	Description	Reference
000b	Short block format address descriptor	3.5.5.2
011b	Long block format address descriptor	3.5.5.3
100b	Bytes from index format address descriptor	3.5.5.4
101b	Physical sector format address descriptor	3.5.5.5
110b	Vendor-specific	
All others	Reserved	

3.5.5.2 Short block format address descriptor

A format type of 000b specifies the short block format address descriptor defined in table 43.

Table 43. Short block format address descriptor (000b)

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	SHORT BLOCK ADDRESS						
3								(LSB)

SHORT BLOCK ADDRESS field

For the FORMAT UNIT command, the SHORT BLOCK ADDRESS field contains the four-byte LBA of a defect. For the READ DEFECT DATA commands, the SHORT BLOCK ADDRESS field contains a vendor-specific four-byte value. For the Translate Address diagnostic pages, the SHORT BLOCK ADDRESS field contains a four-byte LBA or a vendor-specific four byte value that is greater than the capacity of the medium.

3.5.5.3 Long block format address descriptor

A format type of 011b specifies the long block format address descriptor defined in table 44.

Table 44. Long block format address descriptor (011b)

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	LONG BLOCK ADDRESS						
7								(LSB)

LONG BLOCK ADDRESS field

For the FORMAT UNIT command, the LONG BLOCK ADDRESS field contains the eight-byte logical block address of a defect. If multiple logical blocks are contained within a physical block, then the device server may consider logical blocks in addition to the one specified by this descriptor as containing defects. For the READ DEFECT DATA commands, the LONG BLOCK ADDRESS field contains a vendor-specific eight-byte value. For the Translate Address diagnostic pages, the LONG BLOCK ADDRESS field contains a four-byte LBA or a vendor-specific four byte value that is greater than the capacity of the medium.

3.5.5.4 Bytes from index format address descriptor

A format type of 100b specifies the bytes from index address descriptor defined in table 45. For the FORMAT UNIT command and the READ DEFECT DATA commands, this descriptor specifies the location of a defect that is either the length of one track or is no more than eight bytes long. For the Translate Address diagnostic pages, this descriptor specifies the location of a track or the first byte or last byte of an area.

Table 45. Bytes from index format address descriptor (100b)

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	CYLINDER NUMBER						
2								(LSB)
3	HEAD NUMBER							
4	(MSB)	BYTES FROM INDEX						
7								(LSB)

CYLINDER NUMBER field

The CYLINDER NUMBER field contains the cylinder number.

HEAD NUMBER field

The HEAD NUMBER field contains the head number.

BYTES FROM INDEX field

The BYTES FROM INDEX field contains the number of bytes from the index (e.g., from the start of the track) to the location being described. A BYTES FROM INDEX field set to FFFFFFFFh specifies that the entire track is being described.

For sorting bytes from index format address descriptors, the cylinder number is the most significant part of the address and the bytes from index is the least significant part of the address. More than one logical block may be described by this descriptor.

3.5.5.5 Physical sector format address descriptor

A format type of 101b specifies the physical sector address descriptor defined in table 46. For the FORMAT UNIT command and the READ DEFECT DATA commands, this descriptor specifies the location of a defect that is either the length of one track or the length of one sector. For the Translate Address diagnostic pages, this descriptor specifies the location of a track or a sector.

Table 46. Physical sector format address descriptor (101b)

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
	CYLINDER NUMBER							
2								(LSB)
3	HEAD NUMBER							
4	(MSB)							
	SECTOR NUMBER							
7								(LSB)

CYLINDER NUMBER field

The CYLINDER NUMBER field contains the cylinder number.

HEAD NUMBER field

The HEAD NUMBER field contains the head number.

SECTOR NUMBER field

The SECTOR NUMBER field contains the sector number. A SECTOR NUMBER field set to FFFFFFFFh specifies that the entire track is being described.

For sorting physical sector format address descriptors, the cylinder number is the most significant part of the address and the sector number is the least significant part of the address. More than one logical block may be described by this descriptor.

3.6 INQUIRY command

3.6.1 INQUIRY command introduction

The INQUIRY command (see table 47) requests that information regarding the logical unit and SCSI target device be sent to the application client.

Table 47. INQUIRY command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (12h)							
1	Reserved						Obsolete Formerly CMDDT	EVPD
2	PAGE CODE							
3	(MSB)	ALLOCATION LENGTH						
4								(LSB)
5	CONTROL							

EVPD (Enable Vital Product Data) bit

An enable vital product data (EVPD) bit set to one specifies that the device server shall return the vital product data specified by the PAGE CODE field (see 3.6.4).

0 If the EVPD bit is set to zero, the device server shall return the standard INQUIRY data (see 3.6.2). If the PAGE CODE field is not set to zero when the EVPD bit is set to zero, the 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.

1 When the EVPD bit is set to one, the PAGE CODE field specifies which page of vital product data information the device server shall return (see 4.4).

CMDDT (Command Support Data) bit

This bit has been declared Obsolete by T10. See SPC-2 for a description of this bit.

If both the EVPD and CMDDT bits are one, the target shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and an additional sense code of Invalid Field in CDB. When the EVPD bit is one, the Page or OPERATION CODE field specifies which page of vital product data information the target shall return.

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6. If EVPD is set to zero, the allocation length should be at least five, so that the ADDITIONAL LENGTH field in the parameter data (see 3.6.2) is returned. If EVPD is set to one, the allocation length should be should be at least four, so that the PAGE LENGTH field in the parameter data (see 4.4) is returned.

In response to an INQUIRY command received by an incorrect logical unit, the SCSI target device shall return the INQUIRY data with the peripheral qualifier set to the value defined in 3.6.2. The INQUIRY command shall return CHECK CONDITION status only when the device server is unable to return the requested INQUIRY data.

If an INQUIRY command is received from an initiator port with a pending unit attention condition (i.e., before the device server reports CHECK CONDITION status), the device server shall perform the INQUIRY command and shall not clear the unit attention condition (see SAM-4).

The INQUIRY data should be returned even though the device server is not ready for other commands. The standard INQUIRY data should be available without incurring any media access delays. If the device server does store some of the standard INQUIRY data or VPD data on the media, it may return ASCII spaces (20h) in ASCII fields and zeros in other fields until the data is available from the media.

The INQUIRY data may change as the SCSI target device and its logical units perform their initialization sequence. (E.g., logical units may provide a minimum command set from nonvolatile memory until they load the final firmware from the media. After the firmware has been loaded, more options may be supported and therefore different INQUIRY data may be returned.)

If the INQUIRY data changes for any reason, the device server shall establish a unit attention condition for the initiator port associated with every I_T nexus (see SAM-4), with the additional sense code set to INQUIRY DATA HAS CHANGED.

The INQUIRY command may be used by an application client after a hard reset or power on condition to determine the device types for system configuration.

3.6.2 Standard INQUIRY data

The standard INQUIRY data (see table 48) shall contain at least 36 bytes

Table 48. Standard INQUIRY data format

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	RMB	Reserved						
2	VERSION							
3	Obsolete	Obsolete	NORMACA	HISUP	RESPONSE DATA FORMAT			
4	ADDITIONAL LENGTH (N-4)							
5	SCCS	ACC	TPGS		3PC	Reserved		PROTECT
6	Obsolete Formerly BQUE	ENCSERV	VS	MULTIP	MCHNGR	Obsolete	Obsolete	ADDR16 [a]
7	Obsolete	Obsolete	WBUS16[a]	SYNC [a]	Obsolete Formerly LINKED	Obsolete	CMDQUE	VS
8	(MSB)	T10 VENDOR IDENTIFICATION						
15								(LSB)
16	(MSB)	PRODUCT IDENTIFICATION						
31								(LSB)
32	(MSB)	PRODUCT REVISION LEVEL						
35								(LSB)
36 - 43	DRIVE SERIAL NUMBER							
44 - 55	Vendor Unique Seagate fills this field with 00h.							
56	Reserved				CLOCKING [a]		QAS [a]	IUS [a]
57	Reserved							
58	(MSB)	VERSION DESCRIPTOR 1						
59								(LSB)

Table 48. Standard INQUIRY data format

Bit / Byte	7	6	5	4	3	2	1	0
	. .							
72	(MSB)	VERSION DESCRIPTOR 8						
73								(LSB)
74	Reserved							
95								
	Vendor specific parameters							
96	Copyright Notice (Vendor specific)							
n								

[a] The meanings of these fields are specific to SPI-5 (see 3.6.3). For SCSI transport protocols other than the SCSI Parallel Interface, these fields are reserved.

PERIPHERAL QUALIFIER and PERIPHERAL DEVICE TYPE fields

The PERIPHERAL QUALIFIER field and PERIPHERAL DEVICE TYPE field identify the peripheral device connected to the logical unit. If the SCSI target device is not capable of supporting a peripheral device connected to this logical unit, the device server shall set these fields to 7Fh (i.e., PERIPHERAL QUALIFIER field set to 011b and PERIPHERAL DEVICE TYPE field set to 1Fh).

The peripheral qualifier is defined in table 49 and the peripheral device type is defined in table 50.

Table 49. PERIPHERAL QUALIFIER

Qualifier	Description
000b	A peripheral device having the specified peripheral device type is connected to this logical unit. If the device server is unable to determine whether or not a peripheral device is connected, it also shall use this peripheral qualifier. This peripheral qualifier does not mean that the peripheral device connected to the logical unit is ready for access.
001b	A peripheral device having the specified peripheral device type is not connected to this logical unit. However, the device server is capable of supporting the specified peripheral device type on this logical unit.
010b	Reserved
011b	The device server is not capable of supporting a peripheral device on this logical unit. For this peripheral qualifier the peripheral device type shall be set to 1Fh. All other peripheral device type values are reserved for this peripheral qualifier.
100b - 111b	Vendor specific

Table 50. PERIPHERAL DEVICE TYPE

Code	Doc. [a]	Description
00h	SBC-3	Direct access block device (e.g., magnetic disk)
01h	SSC-3	Sequential-access device (e.g., magnetic tape)
02h	SSC	Printer device
03h	SPC-2	Processor device
04h	SBC	Write-once device (e.g., some optical disks)
05h	MMC-5	CD/DVD device
06h		Scanner device (obsolete)
07h	SBC	Optical memory device (e.g., some optical disks)
08h	SMC-3	Medium changer device (e.g., jukeboxes)
09h		Communications device (obsolete)
0Ah - 0Bh		Obsolete
0Ch	SCC-2	Storage array controller device (e.g., RAID)
0Dh	SES	Enclosure services device
0Eh	RBC	Simplified direct-access device (e.g., magnetic disk)
0Fh	OCRW	Optical card reader/writer device
10h	BCC	Bridge Controller Commands
11h	OSD	Object-based Storage Device
12h	ADC-2	Automation/Drive Interface
13h - 1Dh		Reserved
1Eh		Well known logical unit [b]
1Fh		Unknown or no device type

[a] All standards are subject to revision, and parties to agreements based on the standard are encouraged to investigate the possibility of applying the most recent editions of the listed standards.

[b] All well known logical units use the same peripheral device type code.

RMB (Removable Media) bit

0 A removable medium (RMB) bit set to zero indicates that the medium is not removable.

1 A RMB bit set to one indicates that the medium is removable.

VERSION field

The VERSION field indicates the implemented version of the SPC standard and is defined in table 51.

Table 51. VERSION field codes

Code	Description		
00h	The device does not claim conformance to any standard.		
02h	Obsolete		
03h	The device complies to ANSI INCITS 301-1997 (SPC).		
04h	The device complies to ANSI INCITS 351-2001 (SPC-2).		
05h	The device complies to ANSI INCITS 408-2005 (SPC-4).		
06h	The device complies to T10/1731-D (SPC-4)		
Code	**Description**	**Code**	**Description**
01h	Obsolete (SCSI=001b)	06h - 07h	Reserved
08h - 0Ch	Obsolete (ECMA=001b)	0Dh - 3Fh	Reserved
40h - 44h	Obsolete (ISO=01b)	45h - 47h	Reserved
48h - 4Ch	Obsolete (ISO=01b & ECMA=001b)	4Dh - 7Fh	Reserved
80h - 84h	Obsolete (ISO=10b)	85h - 87h	Reserved
88h - 8Ch	Obsolete (ECMA=001b)	8Dh - FFh	Reserved

NORMACA (Normal ACA Supported

1 The Normal ACA Supported (NORMACA) bit set to one indicates that the device server supports a NACA bit set to one in the CDB CONTROL byte and supports the ACA task attribute (see SAM-4).

0 A NORMACA bit set to zero indicates that the device server does not support a NACA bit set to one and does not support the ACA task attribute.

HISUP (Hierarchical Support) bit

0 A hierarchical support (HISUP) bit set to zero indicates the SCSI target device does not use the hierarchical addressing model to assign LUNs to logical units.

1 A HISUP bit set to one indicates the SCSI target device uses the hierarchical addressing model to assign LUNs to logical units.

RESPONSE DATA FORMAT field

A RESPONSE DATA FORMAT field value of two indicates that the data shall be in the format defined in this manual. Response data format values less than two are obsolete. Response data format values greater than two are reserved.

ADDITIONAL LENGTH field

The ADDITIONAL LENGTH field indicates the length in bytes of the remaining standard INQUIRY data. The relationship between the ADDITIONAL LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

SCCS (SCC Supported) bit

1 An SCC Supported (SCCS) bit set to one indicates that the SCSI target device contains an embedded storage array controller component. See SCC-2 for details about storage array controller devices.

0 An SCCS bit set to zero indicates that the SCSI target device does not contain an embedded storage array controller component.

ACC (Access Controls Coordinator) bit

1 An Access Controls Coordinator (ACC) bit set to one indicates that the SCSI target device contains an access controls coordinator that may be addressed through this logical unit.

0 An ACC bit set to zero indicates that no access controls coordinator may be addressed through this logical unit. If the SCSI target device contains an access controls coordinator that may be addressed through any logical unit other than the ACCESS CONTROLS well known logical unit, then the ACC bit shall be set to one for LUN 0.

TPGS (Target Port Group Support) field

The contents of the target port group support (TPGS) field (see table 52) indicate the support for asymmetric logical unit access.

Table 52. Target Port Group Support codes

Code	Description
00b	The SCSI target device does not support asymmetric logical unit access or supports a form of asymmetric access that is vendor specific. Neither the REPORT TARGET GROUPS nor the SET TARGET GROUPS commands is supported.
01b	Only implicit asymmetric logical unit access (see SPC-4) is supported. The SCSI target device is capable of changing target port asymmetric access states without a SET TARGET PORT GROUPS command. The REPORT TARGET PORT GROUPS command is supported and the SET TARGET PORT GROUPS command is not supported.
10b	Only explicit asymmetric logical unit access (i.e., the TPGS field contains 10b or 11b) is supported. The SCSI target device only changes target port asymmetric access states as requested with the SET TARGET PORT GROUPS command. Both the REPORT TARGET PORT GROUPS command and the SET TARGET PORT GROUPS command are supported.
11b	Both explicit and implicit asymmetric logical unit access are supported. Both the REPORT TARGET PORT GROUPS command and the SET TARGET PORT GROUPS commands are supported.

3PC (Third-Party Copy) bit

1 A Third-Party Copy (3PC) bit set to one indicates that the SCSI target device supports third-party copy commands such as the EXTENDED COPY command.

0 A 3PC bit set to zero indicates that the SCSI target device does not support such commands.

PROTECT bit

0 A PROTECT bit set to zero indicates that the logical unit does not support protection information (i.e. type 0) (see SPC-4 and SBC-3).

1 A PROTECT bit set to one indicates that the logical unit supports type 1 protection, type 2 protection, or type 3 protection (see SBC-3). The SPT field in the Extended INQUIRY Data VPD page (see 4.4.6) indicates which type of protection the logical unit supports.

BQUE (Basic Queuing) bit

This bit has been declared Obsolete by T10.

If the logical unit supports the BQUE bit, the BQUE bit combines with the CMDQUE bit to indicate whether the logical unit supports the full task management model or the basic task management model as described in table 53.

ENCSERV (Enclosure Services) bit

1 An Enclosure Services (ENCSERV) bit set to one indicates that the SCSI target device contains an embedded enclosure services component. See SES for details about enclosure services, including a device model for an embedded enclosure services device.

0 An ENCSERV bit set to zero indicates that the SCSI target device does not contain an embedded enclosure services component.

MULTIP (Multi Port) bit

1 A Multi Port (MULTIP) bit set to one indicates that this is a multi-port (two or more ports) SCSI target device and conforms to the SCSI multi-port device requirements found in the applicable standards (e.g., SAM-4, a SCSI transport protocol standard and possibly provisions of a command standard).

0 A MULTIP bit set to zero indicates that this SCSI target device has a single port and does not implement the multi-port requirements.

MCHNGR (Medium Changer) bit

1 A medium changer (MCHNGR) bit set to one indicates that the SCSI target device supports commands to control an attached media changer. See SMC-3 for details about medium changers, including a device model for an attached medium changer device. The MCHNGR bit is valid only when the RMB bit is equal to one.

0 A MCHNGR bit set to zero indicates that the SCSI target device does not support commands to control an attached media changer.

LINKED (Linked Command) bit

This bit has been declared Obsolete by T10.

1 A linked command (LINKED) bit set to one indicates that the device server supports linked commands (see SAM-4).

0 A LINKED bit set to zero indicates the device server does not support linked commands.

CMDQUE (Command Queuing) bit

If the logical unit does not support the BQUE bit, CMDQUE bit shall be set to one indicating that the logical unit supports the task management model (see SAM-4).

If the logical unit supports the BQUE bit, the CMDQUE bit and BQUE bit indicate whether the logical unit supports the full task management model as described in table 53.

Table 53. BQue and CmdQue bit combinations

BQue	CmdQue	Description
0	0	Obsolete
0	1	Full task management model supported
1	0	Basic task management model supported
1	1	Illegal combination of BQUE and CMDQUE bits

T10 VENDOR IDENTIFICATION field

The T10 VENDOR IDENTIFICATION field contains eight bytes of left-aligned ASCII data (see 4.4.2) identifying the vendor of the product. The T10 vendor identification shall be one assigned by INCITS. Bytes 8 through 15 contain the ASCII characters that represent “SEAGATE” (53h 45h 41h 47h 41h 54h 45h 20h (space)).

PRODUCT IDENTIFICATION

The PRODUCT IDENTIFICATION field contains sixteen bytes of left-aligned ASCII data (see 4.4.2) defined by Seagate. Bytes 16 through 31 indicate the drive model with 20h (space) used as a filler. The table below is an example of drive test data returned by the drive. Bytes 16 and 17 will contain 53 54 for all drive models.

Byte 18 through 26 values that may be returned by a Seagate 15K5 parallel SCSI drive:

Parallel SCSI	53	54	33	33	30	30	36	35	35	4C	57
Fibre Channel (FC)	53	54	33	33	30	30	36	35	35	4C	43
Serial Attached SCSI (SAS)	53	54	33	31	35	36	38	35	35	4C	57
ST92811A	53	54	33	31	35	36	38	35	35	4C	43
ST373455LW	53	54	33	37	33	34	35	35	4C	57	20
ST373455LC	53	54	33	37	33	34	35	35	4C	43	20

Refer to the applicable Seagate product manual to get the actual test string.

PRODUCT REVISION LEVEL field

The PRODUCT REVISION LEVEL field contains four bytes of left-aligned ASCII data defined by Seagate. Bytes 32 through 35 contain the four ASCII digits representing the last four digits of the product Firmware Release number. This information is also given in the Vital Product Data page C0h, together with servo RAM and ROM release numbers.

VERSION DESCRIPTOR fields

The VERSION DESCRIPTOR fields provide for identifying up to eight standards to which the SCSI target device claims conformance. The value in each VERSION DESCRIPTOR field shall be selected from the Version Descriptor Values Table in the current SPC standard. All version descriptor values not listed in table 54 are reserved. Technical Committee T10 of INCITS maintains an electronic copy of the information in the Version Descriptor Values Table on its world wide web site (http://www.t10.org/). In the event that the T10 world wide web site is no longer active, access may be possible via the INCITS world wide web site (http://www.incits.org), the ANSI world wide web site (http://www.ansi.org), the IEC site (http://www.iec.ch/), the ISO site (http://www.iso.ch/), or the ISO/IEC JTC 1 web site (http://www.jtc1.org/). It is recommended that the first version descriptor be used for the SCSI architecture standard, followed by the physical transport standard if any, followed by the SCSI transport protocol standard, followed by the appropriate SPC version, followed by the device type command set, followed by a secondary command set if any. Refer to the latest SPC version for a complete list of descriptor values.

Copyright Notice field

Seagate uses bytes 96 - 143 to provide a copyright notice that begins “Copyright (c) XXXX Seagate All rights reserved,” where “XXXX” indicates the year the drive’s firmware code was built (for example: 2005).

3.6.3 SCSI Parallel Interface specific INQUIRY data

Portions of bytes 6 and 7 and all of byte 56 of the standard INQUIRY data shall be used only by SCSI target devices that implement the SCSI Parallel Interface. These fields are noted in table 48. For details on how the SPI-specific fields relate to the SCSI Parallel Interface see SPI-n (where n is 2 or greater). Table 54 shows just the SPI-specific standard INQUIRY fields. The definitions of the SCSI Parallel Interface specific fields shall be as follows.

Table 54. SPI-specific standard INQUIRY bits

Bit Byte	7	6	5	4	3	2	1	0
6	SEE table 48							ADDR16
7	SEE table 48		WBUS16	SYNC	SEE table 48	Obsolete	SEE table 48	
	. . .							
56	Reserved				CLOCKING		QAS	IUS

ADDR16 (wide SCSI address 16) bit

1 A wide SCSI address 16 (ADDR16) bit of one indicates that the SCSI target device supports 16-bit wide SCSI addresses.

0 A value of zero indicates that the SCSI target device does not support 16-bit wide SCSI addresses.

WBUS16 (wide bus 16) bit

1 A wide bus 16 (WBUS16) bit of one indicates that the SCSI target device supports 16-bit wide data transfers.

0 A value of zero indicates that the SCSI target device does not support 16-bit wide data transfers.

SYNC (synchronous transfer) bit

1 A synchronous transfer (SYNC) bit of one indicates that the SCSI target device supports synchronous data transfer.

0 A value of zero indicates the SCSI target device does not support synchronous data transfer.

The obsolete bit 2 in byte 7 indicates whether the SCSI target device supports an obsolete data transfers management mechanism defined in SPI-2.

Table 55 defines the relationships between the ADDR16 and WBUS16 bits.

Table 55. ADDR and WBUS bit relationships

addr16	wbus16	Description
0	0	8 bit wide data path on a single cable with 8 SCSI IDs supported
0	1	16 bit wide data path on a single cable with 8 SCSI IDs supported
1	1	16 bit wide data path on a single cable with 16 SCSI IDs supported

CLOCKING field

The CLOCKING field shall not apply to asynchronous transfers and is defined in table 56.

Table 56. CLOCKING codes

Code	Description
00b	Indicates the target port supports only ST
01b	Indicates the target port supports only DT
10b	Reserved
11b	Indicates the target port supports ST and DT

QAS (Quick Arbitration And Selection Supported) bit

1 A quick arbitration and selection supported (QAS) bit of one indicates that the target port supports quick arbitration and selection.

0 A value of zero indicates that the target port does not support quick arbitration and selection.

IUS (information units supported) bit

1 An information units supported (IUS) bit of one indicates that the SCSI target device supports information unit transfers.

0 A value of zero indicates that the SCSI target device does not support information unit transfers.

The acronyms ST and DT and the terms 'quick arbitration and selection' and 'information units' are defined in SPI-5.

3.6.4 Vital product data

The application client requests the vital product data information by setting the EVPD bit to one and specifying the page code of a vital product data. See 4.4 for details about vital product data. The information returned consists of configuration data (e.g., vendor identification, product identification, model, serial number), manufacturing data (e.g., plant and date of manufacture), field replaceable unit data and other vendor specific or device specific data. If the device server does not implement the requested page, the 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.

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

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

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

3.7 LOCK-UNLOCK Cache (10) command

Obsolete. If received, a CHECK CONDITION status is sent.

3.8 LOCK-UNLOCK Cache (16) command

Obsolete. If received, a CHECK CONDITION status is sent.

3.9 LOG SELECT command

The LOG SELECT command (see table 57) provides a means for an application client to manage statistical information maintained by the SCSI target device about the SCSI target device or its logical units. Device servers that implement the LOG SELECT command shall also implement the LOG SENSE command. Structures in the form of log parameters within log pages are defined as a way to manage the log data. The LOG SELECT command provides for sending zero or more log pages via the Data-Out Buffer. This manual defines the format of the log pages (see 4.2.1), but does not define the conditions and events that are logged. Application clients should issue LOG SENSE commands prior to issuing LOG SELECT commands to determine supported log pages and page lengths.

Table 57. LOG SELECT command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (4Ch)							
1	Reserved						PCR	SP
2	PC		PAGE CODE					
3	SUBPAGE CODE							
4	Reserved							
5								
6								
7	(MSB)	PARAMETER LIST LENGTH						
8								(LSB)
9	CONTROL							

PCR (Parameter Code Reset) bit

The parameter code reset (PCR) bit instructs a device server whether or not to set parameters to their vendor specific default values (e.g., zero) as described in table 60.

SP (Save Parameters) bit

The save parameters (SP) bit instructs a device server whether or not to save parameters to non-volatile memory as described in table 60.

PC (Page Control) field

The page control (PC) field specifies which data counter parameter values (i.e., when the FORMAT AND LINKING field (see 4.2.1) contains 00b or 10b) shall be processed by a device server in response to a LOG

SELECT command as described in table 58. The PC field shall be ignored for list parameters (i.e., when the FORMAT AND LINKING field contains 01b or 11b).

Table 58. Page Control (PC) field values

Values	Description
00b	Threshold values
01b	Cumulative values
10b	Default threshold values
11b	Default cumulative values

The threshold values and cumulative values for data counter parameters are:

1) the current values if there has been an update to a cumulative parameter value (e.g., by a LOG SELECT command or by a device specific event) in the specified page or pages since the last logical unit reset occurred;
2) the saved values, if saved parameters are implemented, current values have been saved, and an update has not occurred since the last logical unit reset; or
3) the vendor specific default values, if saved values are not available or not implemented.

When evaluated together, the combination of the values in the PCR bit, the SP bit, and the PC field specify the actions that a SCSI target device performs while processing a LOG SELECT command (see table 60).

PARAMETER LIST LENGTH field

If the PARAMETER LIST LENGTH field is set to zero, the PAGE CODE field and SUBPAGE CODE field specify the log page or log pages to which the other CDB fields apply (see 3.9.1).

Since each log page in the parameter list contains a PAGE CODE field and SUBPAGE CODE field (see 4.2.1), the 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, if:

a) the PARAMETER LIST LENGTH field contains a value other than zero, and:
   A) the PAGE CODE field contains a value other than zero; or
   B) the SUBPAGE CODE field contains a value other than zero.

The PARAMETER LIST LENGTH field specifies the length in bytes of the parameter list that shall be located in the Data-Out Buffer.

If the PARAMETER LIST LENGTH field contains zero, no log pages shall be transferred. This condition shall not be considered an error. The LOG SELECT command shall be processed as described in 3.9.1.

3.9.1 Processing LOG SELECT when the parameter list length is zero

If the PARAMETER LIST LENGTH field is set to zero (i.e., when there is no parameter data being sent with a LOG SELECT command), the SCSI target device responds by processing the log parameter values as described in this subclause.

PAGE CODE field and SUBPAGE CODE field

The PAGE CODE field and SUBPAGE CODE field (see table 59) specify the log page or log pages to which the other CDB fields apply (see table 59).

TABLE 59. PAGE CODE field and SUBPAGE CODE field

PAGE CODE field	SUBPAGE CODE field	Description
00h	00h	All log parameters in all log pages [a]
00h to 3Fh	01h to FEh	All log parameters in the log page specified by the page code and subpage code
00h to 3Fh	FFh	All log parameters in the log pages specified by page code and all subpage codes
01h to 3Fh	00h	All log parameters in the log page specified by the page code

[a] This is equivalent to the LOG SELECT command operation specified by previous versions of this standard.

Table 60 defines the meaning of the combinations of values for the PCR bit, the SP bit, and the PC field.

Table 60. PCR bit, SP bit, and PC field meanings when parameter list length is zero (Sheet 1 of 2)

PCR bit	SP bit	PC field	Description
0b	0b	0xb	This is not an error. The device server shall make no changes to any log parameter values and shall not save any values to non-volatile media.
0b	1b	00b	The device server shall make no changes to any log parameter values and shall process the optional saving of current parameter values as follows: a) if the values are current threshold data counter parameters, then: A) if the device server implements saving of the current threshold values, the device server shall save all current threshold values to non-volatile media; or B) if the device server does not implement saving of the current threshold values, the device server shall terminate the command[a]. C) or b) if the values are current list parameters, then: A) if the device server implements saving of the current list parameters, the device server shall save all current list parameters to non-volatile media; or B) if the device server does not implement saving of the current list parameters, the device server shall terminate the command[a].

[a] The 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.
[b] Vendor specific default threshold values and vendor specific default cumulative values may be zero.

Table 60. PCR bit, SP bit, and PC field meanings when parameter list length is zero (Sheet 2 of 2)

PCR bit	SP bit	PC field	Description
0b	1b	01b	The device server shall make no change to any log parameter values and shall process the optional saving of current parameter values as follows: a) if the values are current cumulative data counter parameters, then: A) if the device server implements saving of the current cumulative values, the device server shall save all current cumulative values to non-volatile media; or B) if the device server does not implement saving of the current cumulative values, the device server shall terminate the command[a]. C) or b) if the values are current list parameters, then: A) if the device server implements saving of the current list parameters, the device server shall save all current list parameters to non-volatile media; or B) if the device server does not implement saving of the current list parameters, the device server shall terminate the command[a].
0b	xb	10b	The device server shall set all current threshold values to the vendor specific default threshold values[b] and shall not save any values to non-volatile media.
0b	xb	11b	The device server shall set all current cumulative values to the vendor specific default cumulative values[b] and shall not save any values to non-volatile media.
1b	0b	xxb	The device server shall: a) set all current threshold values to the vendor specific default threshold values[b]; b) set all current cumulative values to the vendor specific default cumulative values[b]; c) set all list parameters to their vendor specific default values; and d) not save any values to non-volatile media.
1b	1b	00b	The device server shall process the optional saving of current threshold values as follows: a) if the device server implements saving of the current threshold values, the device server shall: 1) save all current threshold values to non-volatile media; 2) set all current threshold values to the vendor specific default threshold values[b]; 3) set all current cumulative values to the vendor specific default cumulative values[b], and 4) set all list parameters to their vendor specific default values. 5) or b) if the device server does not implement saving of the current threshold values, the device server shall terminate the command[a].
1b	1b	01b	The device server shall process the optional saving of current cumulative values as follows: a) if the device server implements saving of the current cumulative values, the device server shall: 1) save all current cumulative values to non-volatile media; 2) set all current threshold values to the vendor specific default threshold values[b]; 3) set all current cumulative values to the vendor specific default cumulative values[b], and 4) set all list parameters to their vendor specific default values. 5) or b) if the device server does not implement saving of the current cumulative values, the device server shall terminate the command[a].
1b	1b	1xb	The device server shall: a) set all current threshold values to the vendor specific default threshold values[b]; b) set all current cumulative values to the vendor specific default cumulative values[b]; c) set all list parameters to their vendor specific default values; and d) not save any values to non-volatile media.

a The 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.

b Vendor specific default threshold values and vendor specific default cumulative values may be zero.

The current cumulative values may be updated by the device server as defined for the specific log page or by the application client using the LOG SELECT command. The current threshold values may only be modified by the application client via the LOG SELECT command.

Note. Log pages or log parameters that are not available may become available at some later time (e.g., after the logical unit has become ready).

Additional information about the LOG SELECT command is in SPC-4.

The application client should send log pages in ascending order by page code value if the Data-Out Buffer contains multiple log pages. If the Data-Out Buffer contains multiple log parameters within a log page, then they should be sent in ascending order by parameter code value. If the application client sends log pages out of order or parameter codes out of order, the 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 PARAMETER LIST.

3.10 LOG SENSE command

The LOG SENSE command (see table 61) 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. It is a complementary command to the LOG SELECT command. See clause 4.2 of this manual for more information on returned log pages.

Table 61. LOG SENSE command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (4Dh)							
1	Reserved						Obsolete	SP
2	PC		PAGE CODE					
3	Reserved							
4								
5	(MSB)	PARAMETER POINTER						
6								(LSB)
7	(MSB)	ALLOCATION LENGTH						
8								(LSB)
9	CONTROl							

SP (Saving Parameters) bit

1 If saving log parameters is implemented, an SP bit set to one specifies that the device server shall perform the specified LOG SENSE command and shall save all log parameters identified as saveable by the DS bit to a nonvolatile, vendor specific location. Saving parameters is an optional function of the LOG SENSE command. If the logical unit does not implement saving log parameters and if the save parameters (SP) bit is set to one, then the 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.

0 An SP bit set to zero specifies the device server shall perform the specified LOG SENSE command and shall not save any log parameters.

For data counter log parameters (i.e., when the FORMAT AND LINKING field in the parameter control byte in the log parameter structure (see 4.2.1.2) contains 00b or 10b), the page control (PC) table 58 specifies which log parameter values are to be returned by a device server in response to a LOG SENSE command.

For list parameters (i.e., when the FORMAT ANDLINKING field in the parameter control byte in the log parameter structure (see 4.2.1.2) contains 01b or 11b), the PC field shall be ignored. If the parameters specified by the PAGE CODE field and SUBPAGE CODE field in the CDB are list parameters, then the parameter values returned by a device server in response to a LOG SENSE command are determined as follows:

1) the current list parameter values, if there has been an update to a list parameter value (e.g., by a LOG SELECT command or by a device specific event) in the specified page or pages since the last logical unit reset occurred;
2) the saved list parameter values, if saved parameters are implemented and an update has not occurred since the last logical unit reset; or
3) the vendor specific default list parameter values, if saved values are not available or not implemented and an update has not occurred since the last logical unit reset.

PAGE CODE Field

The PAGE CODE field SUBPAGE CODE field specify which log page of data is being requested. If the log page code and subpage combination is reserved or not implemented, the 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.

PARAMETER POINTER field

The PARAMETER POINTER field allows the application client to request parameter data beginning from a specific parameter code to the maximum allocation length or the maximum parameter code supported by the logical unit, whichever is less. If the value of the PARAMETER POINTER field is larger than the largest available parameter code known to the device server for the specified log page, the 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.

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6.

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

Additional information about the LOG SENSE command is in 4.2.

3.11 MODE SELECT(6) command

The MODE SELECT(6) command (see table 62) provides a means for the application client to specify medium, logical unit, or peripheral device parameters to the device server. Device servers that implement the MODE SELECT(6) command shall also implement the MODE SENSE(6) command. Application clients should issue MODE SENSE(6) prior to each MODE SELECT(6) to determine supported mode pages, page lengths, and other parameters.

Table 62. MODE SELECT(6) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (15h)							
1	Reserved			PF	Reserved			SP
2	Reserved							
3								
4	PARAMETER LIST LENGTH							
5	CONTROL							

Logical units shall share mode parameter header and block descriptor values across all I_T nexuses. I_T nexus loss shall not affect mode parameter header, block descriptor, and mode page values.

Logical units shall maintain current and saved values of each mode page based on any of the policies listed in table 63. The mode page policy used for each mode page may be reported in the Mode Page Policy VPD page (see 4.4.10).

Table 63. Mode page policies

Mode page policy	Number of mode page copies
Shared	One copy of the mode page that is shared by all I_T nexuses.
Per target port	A separate copy of the mode page for each target port with each copy shared by all initiator ports.
Per I_T nexus	A separate copy of the mode page for each I_T nexus

After a logical unit reset, each mode parameter header, block descriptor, and mode page shall revert to saved values if supported or default values if saved values are not supported.

If an application client sends a MODE SELECT command that changes any parameters applying to other I_T nexuses, the device server shall establish a unit attention (see SAM-4) condition for the initiator port associated with every I_T nexus except the I_T nexus on which the MODE SELECT command was received, with the additional sense code set to MODE PARAMETERS CHANGED.

PF (Page Format) bit

0 A page format (PF) bit set to zero specifies that all parameters after the block descriptors are vendor specific.

1 A PF bit set to one specifies that the MODE SELECT parameters following the header and block descriptor(s) are structured as pages of related parameters and are as defined in this manual.

SP (save pages) bit

0 A save pages (SP) bit set to zero specifies that the device server shall perform the specified MODE SELECT operation, and shall not save any mode pages. If the logical unit implements no distinction between current and saved mode pages and the SP bit is set to zero, the 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. An SP bit set to one specifies that the device server shall perform the specified MODE SELECT operation, and shall save to a nonvolatile vendor specific location all the saveable mode pages including any sent in the Data-Out Buffer. Mode pages that are saved are specified by the parameter saveable (PS) bit that is returned in the first byte of each mode page by the MODE SENSE command.

1 If the PS bit is set to one in the MODE SENSE data, then the mode page shall be saveable by issuing a MODE SELECT command with the SP bit set to one. If the logical unit does not implement saved mode pages and the SP bit is set to one, then the 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.

PARAMETER LIST LENGTH field

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

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

The mode parameter list for the MODE SELECT and MODE SENSE commands is defined in 3.13. Parts of each mode parameter list are defined in a device-type dependent manner. Definitions for the parts of each mode parameter list that are unique for each device-type may be found in the applicable command standards.

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

a) If the application client sets any field that is reported as not changeable by the device server to a value other than its current value;

b) If the application client sets any field in the mode parameter header or block descriptor(s) to an unsupported value;

c) If an application client sends a mode page with a page length not equal to the page length returned by the MODE SENSE command for that mode page;

d) If the application client sends an unsupported value for a mode parameter and rounding is not implemented for that mode parameter; or

e) If the application client sets any reserved field in the mode parameter list to a non-zero value and the device server checks reserved fields.

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

a) Rounding the parameter to an acceptable value and terminating the command as described in 2.3; or

b) Terminating 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.

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

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

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

3.12 MODE SELECT(10) command

The MODE SELECT(10) command (see table 64) provides a means for the application client to specify medium, logical unit, or peripheral device parameters to the device server. See the MODE SELECT(6) command (see 3.11) for a description of the fields and operation of this command. Application clients should issue MODE SENSE(10) prior to each MODE SELECT(10) to determine supported mode pages, page lengths, and other parameters. Device servers that implement the MODE SELECT(10) command shall also implement the MODE SENSE(10) command.

Table 64. MODE SELECT(10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (55h)							
1	Reserved			PF	Reserved			SP
2	Reserved							
3								
4								
5								
6								
7	(MSB)	PARAMETER LIST LENGTH						
8								(LSB)
9	CONTROL							

3.13 MODE SENSE(6) command

3.13.1 MODE SENSE(6) command introduction

The MODE SENSE(6) command (see table 65) provides a means for a device server to report parameters to an application client. It is a complementary command to the MODE SELECT(6) command. Device servers that implement the MODE SENSE(6) command shall also implement the MODE SELECT(6) command.

Table 65. MODE SENSE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (1Ah)							
1	Reserved				DBD	Reserved		
2	PC		PAGE CODE					
3	SUBPAGE CODE							
4	ALLOCATION LENGTH							
5	CONTROL							

DBD (disable block descriptors) bit

0 A disable block descriptors (DBD) bit set to zero specifies that the device server may return zero or more block descriptors in the returned MODE SENSE data.

1 A DBD bit set to one specifies that the device server shall not return any block descriptors in the returned MODE SENSE data.

PC (Page Control) field

The page control (PC) field specifies the type of mode parameter values to be returned in the mode pages. The PC field is defined in table 66.

Table 66. Page control (PC) field

Code	Type of parameter	Reference
00b	Current values	3.13.1.1
01b	Changeable values	3.13.1.2
10b	Default values	3.13.1.3
11b	Saved values	3.13.1.4

The PC field only affects the mode parameters within the mode pages, however the PS bit, SPF bit, PAGE CODE field, SUBPAGE CODE field, and PAGE LENGTH field should return current values (i.e., as if PC is set to 00b). The mode parameter header and mode parameter block descriptor should return current values.

Some SCSI target devices may not distinguish between current and saved mode parameters and report identical values in response to a PC field of either 00b or 11b. See also the description of the save pages (SP) bit in the MODE SELECT command.

PAGE CODE and SUBPAGE CODE fields

The PAGE CODE and SUBPAGE CODE fields specify which mode pages and subpages to return (see table 67).

Table 67. Mode page code usage for all devices

Page Code	Subpage Code	Description
00h	vendor specific	Vendor specific (does not require page format)
01h - 1Fh	00h	See specific device types (page_0 format)
	01h - DFh	See specific device types (sub_page format)
	E0h - FEh	Vendor specific (sub_page format)
	FFh	Return all subpages for the specified device specific mode page in the page_0 format for subpage 00h and in the sub_page format for subpages 01h - FEh
20h - 3Eh	00h	Vendor specific (page_0 format required)
	01h - FEh	Vendor specific (sub_page format required)
	FFh	Return all subpages for the specified vendor specific mode page in the page_0 format for subpage 00h and in the sub_page format for subpages 01h - FEh
3Fh	00h	Return all subpage 00h mode pages in page_0 format
	01h - FEh	Reserved
	FFh	Return all subpages for all mode pages in the page_0 format for subpage 00h and in the sub_page format for subpages 01h - FEh

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6.

An application client may request any one or all of the supported mode pages from the device server. If an application client issues a MODE SENSE command with a page code or subpage code value not implemented by the logical unit, the 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.

If an application client requests all supported mode pages, the device server shall return the supported pages in ascending page code order beginning with mode page 01h. If mode page 00h is implemented, the device server shall return mode page 00h after all other mode pages have been returned.

If the PC field and the PAGE CODE field are both set to zero, the device server should return a mode parameter header and block descriptor, if applicable.

The mode parameter list for all device types for MODE SELECT and MODE SENSE is defined in 3.13. Parts of the mode parameter list are specifically defined for each device type. Definitions for the parts of each mode parameter list that are unique for each device-type may be found in the applicable command standards.

3.13.1.1 Current values

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

a) The current values of the mode parameters established by the last successful MODE SELECT command;

b) The saved values of the mode parameters if a MODE SELECT command has not successfully completed since the mode parameters were restored to their saved values (see 3.9); or

c) The default values of the mode parameters if a MODE SELECT command has not successfully completed since the mode parameters were restored to their default values (see 3.9).

3.13.1.2 Changeable values

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

If the logical unit does not implement changeable parameters mode pages and the device server receives a MODE SENSE command with 01b in the PC field, then the 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.

An attempt to change a non-changeable mode parameter using the MODE SELECT command shall result in an error condition (see 3.9).

The application client should issue a MODE SENSE command with the PC field set to 01b and the PAGE CODE field set to 3Fh to determine which mode pages are supported, which mode parameters within the mode pages are changeable, and the supported length of each mode page prior to issuing any MODE SELECT commands.

3.13.1.3 Default values

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

3.13.1.4 Saved values

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

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

3.13.1.5 Initial responses

After a logical unit reset, the device server shall respond in the following manner:

a) If default values are requested, report the default values;

(b) If saved values are requested, report valid restored mode parameters, or restore the mode parameters and report them. If the saved values of the mode parameters are not able to be accessed from the nonvolatile vendor specific location, the command shall be terminated with CHECK CONDITION status, with the sense key set to NOT READY. If saved parameters are not implemented, respond as defined in 3.13.1.4; or

(c) If current values are requested and the current values have been sent by the application client via a MODE SELECT command, the current values shall be returned. If the current values have not been sent, the device server shall return:

   a) The saved values, if saving is implemented and saved values are available; or

   b) The default values.

3.14 MODE SENSE(10) command

The MODE SENSE(10) command (see table 68) provides a means for a device server to report parameters to an application client. It is a complementary command to the MODE SELECT(10) command. Device servers that implement the MODE SENSE(10) command shall also implement the MODE SELECT(10) command.

Table 68. MODE SENSE(10) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (5Ah)							
1	Reserved			LLBAA	DBD	Reserved		
2	PC		PAGE CODE					
3	SUBPAGE CODE							
4	Reserved							
6								
7	(MSB)	ALLOCATION LENGTH						
8								(LSB)
9	CONTROL							

LLBAA (Long LBA Accepted) bit

1 If the Long LBA Accepted (LLBAA) bit is set to one, the device server is allowed to return parameter data with the LONGLBA bit equal to one.

0 If LLBAA bit is set to zero, the LONGLBA bit shall be zero in the parameter data returned by the device server.

See the MODE SENSE(6) command (3.13) for a description of the other fields and operation of this command.

3.15 PERSISTENT RESERVE IN command

3.15.1 PERSISTENT RESERVE IN command introduction

The PERSISTENT RESERVE IN command (see table 69) is used to obtain information about persistent reservations and reservation keys (i.e., registrations) that are active within a device server. This command is used in conjunction with the PERSISTENT RESERVE OUT command (see 3.16).

Table 69. PERSISTENT RESERVE IN command

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="8">OPERATION CODE (5Eh)</td></tr>
<tr><td>1</td><td colspan="3">Reserved</td><td colspan="5">SERVICE ACTION</td></tr>
<tr><td>2
…
6</td><td colspan="8">Reserved</td></tr>
<tr><td>7
8</td><td colspan="8">(MSB)
ALLOCATION LENGTH
(LSB)</td></tr>
<tr><td>9</td><td colspan="8">CONTROL</td></tr>
</table>

The service action codes for the PERSISTENT RESERVE IN command are defined in table 70.

Table 70. PERSISTENT RESERVE IN service action codes

Code	Name	Description	Reference
00h	READ KEYS	Reads all registered reservation keys (i.e., registrations) as described in SPC-4.	3.15.2
01h	READ RESERVATION	Reads the current persistent reservations as described in SPC-4.	3.15.3
02h	REPORT CAPABILITIES	Returns capability information	3.15.4
03h	READ FULL STATUS	Reads complete information about all registrations and the persistent reservations, if any	3.15.5
04h - 1Fh	Reserved	Reserved	

3.15.2 READ KEYS service action

The READ KEYS service action requests that the device server return a parameter list containing a header and a list of each currently registered I_T nexus' reservation key. If multiple I_T nexuses have registered with the same key, then that key value shall be listed multiple times, once for each such registration.

For more information on READ KEYS see SPC-4.

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the READ KEYS service action is shown in table 71.

Table 71. PERSISTENT RESERVE IN parameter data for READ KEYS

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3				PRGENERATION				(LSB)
4	(MSB)							
7				ADDITIONAL LENGTH (N-7)				(LSB)
				Reservation key list				
8	(MSB)							
15				First reservation key				(LSB)
				. . .				
n-7	(MSB)							
n				Last reservation key				(LSB)

PRGENERATION (Persistent Reservations Generation) field

The Persistent Reservations Generation (PRGENERATION) field shall contain a 32-bit counter maintained by the device server that shall be incremented every time a PERSISTENT RESERVE OUT command requests a REGISTER service action, a REGISTER AND IGNORE EXISTING KEY service action, a REGISTER AND MOVE service action, a CLEAR service action, a PREEMPT service action, or a PREEMPT AND ABORT service action. The counter shall not be incremented by a PERSISTENT RESERVE IN command, by a PERSISTENT RESERVE OUT command that performs a RESERVE or RELEASE service action, or by a PERSISTENT RESERVE OUT command that is terminated due to an error or reservation conflict. Regardless of the APTPL bit value the PRGENERATION value shall be set to zero by a power on.

ADDITIONAL LENGTH field

The ADDITIONAL LENGTH field contains a count of the number of bytes in the Reservation key list. The relationship between the ADDITIONAL LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

The reservation key list contains the 8-byte reservation keys for all I_T nexuses that have been registered see SPC-4.

3.15.3 READ RESERVATION service action

3.15.3.1 READ RESERVATION service action introduction

The READ RESERVATION service action requests that the device server return a parameter list containing a header and the persistent reservation, if any, that is present in the device server.

For more information on READ RESERVATION see SPC-4.

3.15.3.2 PERSISTENT RESERVE IN parameter data for READ RESERVATION

When no persistent reservation is held, the format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the READ RESERVATION service action is shown in table 72.

Table 72. PERSISTENT RESERVE IN data for READ RESERVATION with no reservation held

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
...				PRGENERATION				
3								(LSB)
4	(MSB)							
...				ADDITIONAL LENGTH (0)				
7								(LSB)

PRGENERATION field

The PRGENERATION field shall be as defined for the PERSISTENT RESERVE IN command with READ KEYS service action parameter data (see 3.15.2).

ADDITIONAL LENGTH field

The ADDITIONAL LENGTH field shall be set to zero, indicating that no persistent reservation is held.

When a persistent reservation is held, the format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the READ RESERVATION service action is shown in table 73.

PRGENERATION field

The PRGENERATION field shall be as defined for the PERSISTENT RESERVE IN command with READ KEYS service action parameter data.

ADDITIONAL LENGTH field

The ADDITIONAL LENGTH field contains a count of the number of bytes to follow and shall be set to 16. The relationship between the ADDITIONAL LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

RESERVATION KEY field

The RESERVATION KEY field shall contain the reservation key under which the persistent reservation is held (see SPC-4).

SCOPE field

The SCOPE field shall be set to LU_SCOPE (see 3.15.3.3).

TYPE field

The TYPE field shall contain the persistent reservation type (see 3.15.3.4) specified in the PERSISTENT RESERVE OUT command that created the persistent reservation.

The obsolete fields in bytes 16 through 19, byte 22, and byte 23 were defined in a previous standard.

Table 73. PERSISTENT RESERVE IN parameter data for READ RESERVATION with reservation

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				PRGENERATION				
3								(LSB)
4	(MSB)							
				ADDITIONAL LENGTH (10h)				
7								(LSB)
8	(MSB)							
				RESERVATION KEY				
15								(LSB)
16								
				Obsolete				
19								
20				Reserved				
21			SCOPE				TYPE	
22								
				Obsolete				
23								

3.15.3.3 Persistent reservations scope

The SCOPE field (see table 74) shall be set to LU_SCOPE, specifying that the persistent reservation applies to the entire logical unit.

Table 74. Persistent reservation scope codes

Code	Name	Description
0h	LU_SCOPE	Persistent reservation applies to the full logical unit
1h - 2h		Obsolete
3h - Fh		Reserved

The LU_SCOPE scope shall be implemented by all device servers that implement PERSISTENT RESERVE OUT.

3.15.3.4 Persistent reservations type

The TYPE field (see table 75) specifies the characteristics of the persistent reservation being established for all logical blocks within the logical unit. SPC-4 defines the persistent reservation types under which each command defined in that standard is allowed to be processed. Each other command standard (e.g., SBC-3, SSC-3, SMC-3, MMC-5, or SES-2) defines the persistent reservation types under which each command defined in that command standard is allowed to be processed.

Table 75. Persistent reservation type codes

Code	Name	Description
0h		Obsolete
1h	Write Exclusive	**Access Restrictions**: Some commands (e.g., media-access write commands) are only allowed for the persistent reservation holder (see SPC-4). **Persistent Reservation Holder:** There is only one persistent reservation holder.
2h		Obsolete
3h	Exclusive Access	**Access Restrictions**: Some commands (e.g., media-access commands) are only allowed for the persistent reservation holder (see SPC-4). **Persistent Reservation Holder:** There is only one persistent reservation holder.
4h		Obsolete
5h	Write Exclusive – Registrants Only	**Access Restrictions**: Some commands (e.g., media-access write commands) are only allowed for registered I_T nexuses. **Persistent Reservation Holder:** There is only one persistent reservation holder (see SPC-4).
6h	Exclusive Access – Registrants Only	**Access Restrictions**: Some commands (e.g., media-access commands) are only allowed for registered I_T nexuses. **Persistent Reservation Holder:** There is only one persistent reservation holder (see SPC-4).
7h	Write Exclusive – All Registrants	**Access Restrictions**: Some commands (e.g., media-access write commands) are only allowed for registered I_T nexuses. **Persistent Reservation Holder:** Each registered I_T nexus is a persistent reservation holder (see SPC-4).
8h	Exclusive Access – All Registrants	**Access Restrictions**: Some commands (e.g., media-access commands) are only allowed for registered I_T nexuses. **Persistent Reservation Holder:** Each registered I_T nexus is a persistent reservation holder (see SPC-4).
9h - Fh	Reserved	

3.15.4 REPORT CAPABILITIES service action

The REPORT CAPABILITIES service action requests that the device server return information on persistent reservation features.

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the REPORT CAPABILITIES service action is shown in table 76.

Table 76. PERSISTENT RESERVE IN parameter data for REPORT CAPABILITIES

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	LENGTH (0008h)						
7								(LSB)
8	Reserved			CRH	SIP_C	ATP_C	Reserved	PTPL_C
9	TMV	ALLOW COMMANDS			Reserved			PTPL_A
10	PERSISTENT RESERVATION TYPE MASK							
11								
12	Reserved							
13								

LENGTH field

The LENGTH field indicates the length in bytes of the parameter data. The relationship between the LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

CRH (Compatible Reservation Handling) field

1 A Compatible Reservation Handling (CRH) bit set to one indicates that the device server supports the exceptions to the RESERVE and RELEASE commands described in (see SPC-4).

0 A CRH bit set to zero indicates that RESERVE(6) command, RESERVE(10) command, RELEASE(6) command, and RELEASE(10) command are processed as defined in SPC-4.

SIP_C (Specify Initiator Ports Capable) bit

1 A Specify Initiator Ports Capable (SIP_C) bit set to one indicates that the device server supports the SPEC_I_PT bit in the PERSISTENT RESERVE OUT command parameter data (see 3.16.3).

0 An SIP_C bit set to zero indicates that the device server does not support the SPEC_I_PT bit in the PERSISTENT RESERVE OUT command parameter data.

ATP_C (Target Ports Capable) bit

1 An All Target Ports Capable (ATP_C) bit set to one indicates that the device server supports the ALL_TG_PT bit in the PERSISTENT RESERVE OUT command parameter data.

0 An ATP_C bit set to zero indicates that the device server does not support the ALL_TG_PT bit in the PERSISTENT RESERVE OUT command parameter data.

PTPL_C (Persist Through Power Loss Capable) bit

1 A Persist Through Power Loss Capable (PTPL_C) bit set to one indicates that the device server supports the persist through power loss capability see SPC-4 for persistent reservations and the APTPL bit in the PERSISTENT RESERVE OUT command parameter data.

0 An PTPL_C bit set to zero indicates that the device server does not support the persist through power loss capability.

TMV (Type Mask Valid) bit

1 A Type Mask Valid (TMV) bit set to one indicates that the PERSISTENT RESERVATION TYPE MASK field contains a bit map indicating which persistent reservation types are supported by the device server.

0 A TMV bit set to zero indicates that the PERSISTENT RESERVATION TYPE MASK field shall be ignored.

The ALLOW COMMANDS (see table 77) indicates whether certain commands are allowed through certain types of persistent reservations.

Table 77. ALLOW COMMANDS field

Code	Description
000b	No information is provided about whether certain commands are allowed through certain types of persistent reservations.
001b	The device server allows the TEST UNIT READY command (see 3.55) through Write Exclusive and Exclusive Access persistent reservations and does not provide information about whether the following commands are allowed through Write Exclusive persistent reservations: a) the MODE SENSE, READ ATTRIBUTE, READ BUFFER, RECEIVE DIAGNOSTIC RESULTS, REPORT SUPPORTED OPERATION CODES, and REPORT SUPPORTED TASK MANAGEMENT FUNCTION commands (see 3.39); and b) the READ DEFECT DATA command (see SBC-3).
010b	The device server allows the TEST UNIT READY command through Write Exclusive and Exclusive Access persistent reservations and does not allow the following commands through Write Exclusive persistent reservations: a) the MODE SENSE, READ ATTRIBUTE, READ BUFFER, RECEIVE DIAGNOSTIC RESULTS, REPORT SUPPORTED OPERATION CODES, and REPORT SUPPORTED TASK MANAGEMENT FUNCTION commands; and b) the READ DEFECT DATA command.
011b	The device server allows the TEST UNIT READY command through Write Exclusive and Exclusive Access persistent reservations and allows the following commands through Write Exclusive persistent reservations: a) the MODE SENSE, READ ATTRIBUTE, READ BUFFER, RECEIVE DIAGNOSTIC RESULTS, REPORT SUPPORTED OPERATION CODES, and REPORT SUPPORTED TASK MANAGEMENT FUNCTION commands; and b) the READ DEFECT DATA command.
100b to 111b	Reserved

PTPL_A (Persist Through Power Loss Activated) bit

1 A Persist Through Power Loss Activated (PTPL_A) bit set to one indicates that the persist through power loss capability is activated (see SPC-4).

0 A PTPL_A bit set to zero indicates that the persist through power loss capability is not activated.

The PERSISTENT RESERVATION TYPE MASK field (see table 78) contains a bit map that indicates the persistent reservation types that are supported by the device server.

Table 78. Persistent Reservation Type Mask format

Bit / Byte	7	6	5	4	3	2	1	0
4	WR_EX_AR	EX_AC_RO	WR_EX_RO	Reserved	EX_AC	Reserved	WR_EX	Reserved
5	Reserved							EX_AC_AR

WR_EX_AR (Write Exclusive – All Registrants) bit

1 A Write Exclusive – All Registrants (WR_EX_AR) bit set to one indicates that the device server supports the Write Exclusive – All Registrants persistent reservation type.

0 An WR_EX_AR bit set to zero indicates that the device server does not support the Write Exclusive – All Registrants persistent reservation type.

EX_AC_RO (Exclusive Access – Registrants Only) bit

1 An Exclusive Access – Registrants Only (EX_AC_RO) bit set to one indicates that the device server supports the Exclusive Access – Registrants Only persistent reservation type.

0 An EX_AC_RO bit set to zero indicates that the device server does not support the Exclusive Access – Registrants Only persistent reservation type.

WR_EX_RO (Write Exclusive – Registrants Only) bit

1 A Write Exclusive – Registrants Only (WR_EX_RO) bit set to one indicates that the device server supports the Write Exclusive – Registrants Only persistent reservation type.

0 An WR_EX_RO bit set to zero indicates that the device server does not support the Write Exclusive – Registrants Only persistent reservation type.

EX_AC (Exclusive Access) bit

1 An Exclusive Access (EX_AC) bit set to one indicates that the device server supports the Exclusive Access persistent reservation type.

0 An EX_AC bit set to zero indicates that the device server does not support the Exclusive Access persistent reservation type.

WR_EX (Write Exclusive) bit

1 A Write Exclusive (WR_EX) bit set to one indicates that the device server supports the Write Exclusive persistent reservation type.

0 An WR_EX bit set to zero indicates that the device server does not support the Write Exclusive persistent reservation type.

EX_AC_AR (Exclusive Access – All Registrants)

1 An Exclusive Access – All Registrants (EX_AC_AR) bit set to one indicates that the device server supports the Exclusive Access – All Registrants persistent reservation type.

0 An EX_AC_AR bit set to zero indicates that the device server does not support the Exclusive Access – All Registrants persistent reservation type.

3.15.5 READ FULL STATUS service action

The READ FULL STATUS service action requests that the device server return a parameter list describing the registration and persistent reservation status of each currently registered I_T nexus for the logical unit.

For more information on READ FULL STATUS see SPC-4.

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the READ FULL STATUS service action is shown in table 79.

Table 79. PERSISTENT RESERVE IN parameter data for READ FULL STATUS

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3	PRGENERATION							(LSB)
4	(MSB)							
7	ADDITIONAL LENGTH (N-7)							(LSB)
	Full STatus descriptors							
8	first full status descriptor (see table 80)							
	...							
n	Last full status descriptor (see table 80)							

PRGENERATION field

The PRGENERATION field shall be as defined for the PERSISTENT RESERVE IN command with READ KEYS service action parameter data (see 3.15.2).

ADDITIONAL LENGTH field

The ADDITIONAL LENGTH field contains a count of the number of bytes to follow in the full status descriptors. The relationship between the ADDITIONAL LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

The format of the full status descriptors is shown in table 80. Each full status descriptor describes one or more registered I_T nexuses. The device server shall return persistent reservations status information for every registered I_T nexus.

Table 80. PERSISTENT RESERVE IN full status descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			RESERVATION KEY				
7								(LSB)
8				Reserved				
11								
12				Reserved			ALL_TG_PT	R_HOLDER
13		SCOPE				TYPE		
14				Reserved				
17								
18	(MSB)			RELATIVE TARGET PORT IDENTIFIER				
19								(LSB)
20	(MSB)			ADDITIONAL DESCRIPTOR LENGTH (N-23)				
23								(LSB)
24				TRANSPORTID				
n								

RESERVATION KEY field

The RESERVATION KEY field contains the reservation key.

R_HOLDER (Reservation Holder) bit

1 A Reservation Holder (R_HOLDER) bit set to one indicates that all I_T nexuses described by this full status descriptor are registered and are persistent reservation holders.

0 A R_HOLDER bit set to zero indicates that all I_T nexuses described by this full status descriptor are registered but are not persistent reservation holders.

ALL_TG_PT (All Target Ports) bit

0 An All Target Ports (ALL_TG_PT) bit set to zero indicates that this full status descriptor represents a single I_T nexus.

1 An ALL_TG_PT bit set to one indicates that:

(a) This full status descriptor represents all the I_T nexuses that are associated with both:

a. The initiator port specified by the TRANSPORTID field; and

b. Every target port in the SCSI target device;

(b) All the I_T nexuses are registered with the same reservation key; and

(c) All the I_T nexuses are either reservation holders or not reservation holders as indicated by the R_HOLDER bit.

The device server is not required to return an ALL_TG_PT bit set to one. Instead, it may return separate full status descriptors for each I_T nexus.

R_HOLDER bit

0 If the R_HOLDER bit is set to one (i.e., if the I_T nexus described by this full status descriptor is a reservation holder), the SCOPE field and the TYPE field are as defined in the READ RESERVATION service action parameter data (see 3.15.3).

1 If the R_HOLDER bit is set to zero, the contents of the SCOPE field and the TYPE field are not defined by this manual.

ALL_TG_PT bit

0 If the ALL_TG_PT bit set to zero, the RELATIVE TARGET PORT IDENTIFIER field contains the relative port identifier (see SPC-4) of the target port that is part of the I_T nexus described by this full status descriptor.

1 If the ALL_TG_PT bit is set to one, the contents of the RELATIVE TARGET PORT IDENTIFIER field are not defined by this manual.

ADDITIONAL DESCRIPTOR LENGTH field

The ADDITIONAL DESCRIPTOR LENGTH field contains a count of the number of bytes that follow in the descriptor (i.e., the size of the TRANSPORTID).

TRANSPORTID field

The TRANSPORTID field contains a TRANSPORTID (see SPC-4) identifying the initiator port that is part of the I_T nexus or I_T nexuses described by this full status descriptor.

3.16 PERSISTENT RESERVE OUT command

3.16.1 PERSISTENT RESERVE OUT command introduction

The PERSISTENT RESERVE OUT command (see table 81) is used to request service actions that reserve a logical unit for the exclusive or shared use of a particular I_T nexus. The command uses other service actions to manage and remove such persistent reservations.

I_T nexuses performing PERSISTENT RESERVE OUT service actions are identified by a registered reservation key provided by the application client. An application client may use the PERSISTENT RESERVE IN command to obtain the reservation key, if any, for the I_T nexus holding a persistent reservation and may use the PERSISTENT RESERVE OUT command to preempt that persistent reservation.

Table 81. PERSISTENT RESERVE OUT command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (5Fh)							
1	Reserved			SERVICE ACTION				
2	SCOPE				TYPE			
3	Reserved							
4								
5	(MSB)	PARAMETER LIST LENGTH						
8								(LSB)
9	CONTROL							

If a PERSISTENT RESERVE OUT command is attempted, but there are insufficient device server resources to complete the operation, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INSUFFICIENT REGISTRATION RESOURCES.

The PERSISTENT RESERVE OUT command contains fields that specify a persistent reservation service action, the intended scope of the persistent reservation, and the restrictions caused by the persistent reservation. The TYPE and SCOPE fields are defined in 3.15.3.4 and 3.15.3.3. If a SCOPE field specifies a scope that is not implemented, the 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.

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

PARAMETER LIST LENGTH field

The PARAMETER LIST LENGTH field specifies the number of bytes of parameter data for the PERSISTENT RESERVE OUT command.

The parameter list shall be 24 bytes in length and the PARAMETER LIST LENGTH field shall contain 24 (18h), if the following conditions are true:

a) The SPEC_I_PT bit (see 3.16.3) is set to zero; and

b) The service action is not REGISTER AND MOVE.

If the SPEC_I_PT bit is set to zero, the service action is not REGISTER AND MOVE, and the parameter list length is not 24, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER LIST LENGTH ERROR.

If the parameter list length is larger than the device server is able to process, the command should be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to PARAMETER LIST LENGTH ERROR.

3.16.2 PERSISTENT RESERVE OUT service actions

When processing the PERSISTENT RESERVE OUT service actions, the device server shall increment the PRGENERATION value as specified in 3.15.2.

The PERSISTENT RESERVE OUT command service actions are defined in table 82.

Table 82. PERSISTENT RESERVE OUT service action codes

Code	Name	Description	PRGENERATION field incremented (see 3.15.2)	Parameter list format
00h	REGISTER	Register a reservation key with the device server (see SPC-4) or unregister a reservation key (see SPC-4).	Yes	Basic (see 3.16.3)
01h	RESERVE	Creates a persistent reservation having a specified SCOPE and TYPE (see SPC-4). The SCOPE and TYPE of a persistent reservation are defined in 3.15.3.3 and 3.15.3.4.	No	Basic (see 3.16.3)
02h	RELEASE	Releases the selected persistent reservation (see SPC-4).	No	Basic (see 3.16.3)
03h	CLEAR	Clears all reservation keys (i.e., registrations) and all persistent reservations (see SPC-4).	Yes	Basic (see 3.16.3)
04h	PREEMPT	Preempts persistent reservations and/or removes registrations (see SPC-4).	Yes	Basic (see 3.16.3)
05h	PREEMPT AND ABORT	Preempts persistent reservations and/or removes registrations and aborts all tasks for all preempted I_T nexuses (see SPC-4)	Yes	Basic (see 3.16.3)
06h	REGISTER AND IGNORE EXISTING KEY	Register a reservation key with the device server (see SPC-4) or unregister a reservation key (see SPC-4).	Yes	Basic (see 3.16.3)
07h	REGISTER AND MOVE	Register a reservation key for another I_T nexus with the device server and move a persistent reservation to that I_T nexus (see SPC-4)	Yes	Register and move (see 3.16.4)
08h - 1Fh	Reserved			

3.16.3 Basic PERSISTENT RESERVE OUT parameter list

The parameter list format shown in table 83 shall be used by the PERSISTENT RESERVE OUT command with any service action except the REGISTER AND MOVE service action. All fields shall be sent, even if the field is not required for the specified service action and scope values.

Table 83. PERSISTENT RESERVE OUT parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			RESERVATION KEY				
7								(LSB)
8	(MSB)			SERVICE ACTION RESERVATION KEY				
15								(LSB)
16				Obsolete				
19								
20	Reserved				SPEC_I_PT	ALL_TG_PT	Reserved	APTPL
21	Reserved							
22				Obsolete				
23								
24				ADDITIONAL PARAMETER DATA				
n								

The obsolete fields in bytes 16 through 19, byte 22 and byte 23 were defined in a previous standard.

RESERVATION KEY field

The RESERVATION KEY field contains an 8-byte value provided by the application client to the device server to identify the I_T nexus that is the source of the PERSISTENT RESERVE OUT command. The device server shall verify that the contents of the RESERVATION KEY field in a PERSISTENT RESERVE OUT command parameter data matches the registered reservation key for the I_T nexus from which the command was received, except for:

a) The REGISTER AND IGNORE EXISTING KEY service action where the RESERVATION KEY field shall be ignored; and

b) The REGISTER service action for an unregistered I_T nexus where the RESERVATION KEY field shall contain zero.

Except as noted above, when a PERSISTENT RESERVE OUT command specifies a RESERVATION KEY field other than the reservation key registered for the I_T nexus the device server shall return a RESERVATION CONFLICT status. Except as noted above, the reservation key of the I_T nexus shall be verified to be correct regardless of the SERVICE ACTION and SCOPE field values.

SERVICE ACTION RESERVATION KEY field

The SERVICE ACTION RESERVATION KEY field contains information needed for the following service actions: REGISTER, REGISTER AND IGNORE EXISTING KEY, PREEMPT, and PREEMPT AND ABORT. The SERVICE ACTION RESERVATION KEY field is ignored for the following service actions: RESERVE, RELEASE, and CLEAR.

For the REGISTER service action and REGISTER AND IGNORE EXISTING KEY service action, the SERVICE ACTION RESERVATION KEY field contains:

a) The new reservation key to be registered in place of the registered reservation key specified in the RESERVATION KEY field; or

b) Zero to unregister the registered reservation key specified in the RESERVATION KEY field.

For the PREEMPT service action and PREEMPT AND ABORT service action, the SERVICE ACTION RESERVATION KEY field contains the reservation key of:

a) The registrations to be removed; and

b) If the SERVICE ACTION RESERVATION KEY field identifies a persistent reservation holder (see SPC-4), persistent reservations that are to be preempted.

SPEC_I_PT (Specify Initiator Ports) bit

0 If the Specify Initiator Ports (SPEC_I_PT) bit is set to zero, the device server shall apply the registration only to the I_T nexus that sent the PERSISTENT RESERVE OUT command.

1 If the SPEC_I_PT bit is set to one for the REGISTER service action or the REGISTER AND IGNORE EXISTING KEY service action, then the additional parameter data shall include a list of transport IDs (see table 84) and the device server shall also apply the registration to the I_T nexus for each initiator port specified by a TRANSPORTID. If a registration fails for any initiator port (e.g., if the logical unit does not have enough resources available to hold the registration information), none of the other registrations shall be made.

Table 84. PERSISTENT RESERVE OUT specify initiator ports additional parameter data

Bit / Byte	7	6	5	4	3	2	1	0
24	TRANSPORTID PARAMETER DATA LENGTH (N - 27)							
27								
	TRANSPORTIDS LIST							
28	FIRST TRANSPORTID							
	.							
	LAST TRANSPORTID							
n								

TRANSPORTID PARAMETER DATA LENGTH field

The TRANSPORTID PARAMETER DATA LENGTH field specifies the number of bytes of TRANSPORTIDs that follow.

The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST:

(a) If the value in the parameter list length field in the CDB does not include all of the additional parameter list bytes specified by the TRANSPORTID PARAMETER DATA LENGTH field; or

(b) If the value in the TRANSPORTID PARAMETER DATA LENGTH field results in the truncation of a TRANSPORTID.

The format of a TRANSPORTID is specified in SPC-4.

ALL_TG_PT (All Target Ports) bit

The All Target Ports (ALL_TG_PT) bit is valid only for the REGISTER service action and the REGISTER AND IGNORE EXISTING KEY service action, and shall be ignored for all other service actions. Support for the ALL_TG_PT bit is optional.

1 If the device server receives a REGISTER service action or a REGISTER AND IGNORE EXISTING KEY service action with the ALL_TG_PT bit set to one, it shall create the specified registration on all target ports in the SCSI target device known to the device server (i.e., as if the same registration request had been received individually through each target port).

0 If the device server receives a REGISTER service action or a REGISTER AND IGNORE EXISTING KEY service action with the ALL_TG_PT bit set to zero, it shall apply the registration only to the target port through which the PERSISTENT RESERVE OUT command was received.

APTPL (Activate Persist Through Power Loss) bit

1 The Activate Persist Through Power Loss (APTPL) bit is valid only for the REGISTER service action and the REGISTER AND IGNORE EXISTING KEY service action, and shall be ignored for all other service actions. Support for an APTPL bit equal to one is optional. If a device server that does not support an APTPL bit set to one receives that value in a REGISTER service action or a REGISTER AND IGNORE EXISTING KEY service action, the 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 PARAMETER LIST. If the last valid APTPL bit value received by the device server is one, the logical unit shall retain any persistent reservation(s) that may be present and all reservation keys (i.e., registrations) for all I_T nexuses even if power is lost and later returned (see SPC-4).

0 If the last valid APTPL bit value received by the device server is zero, the loss of power in the SCSI target device shall release the persistent reservation for the logical unit and remove all registered reservation keys (see SPC-4).

Table 85 summarizes which fields are set by the application client and interpreted by the device server for each service action and scope value.

Table 85. PERSISTENT RESERVE OUT service actions and valid parameters (part 1 of 2)

Service action	Allowed SCOPE	**Parameters** (part 1 of 2)			
		TYPE	**RESERVATION KEY**	**SERVICE ACTION RESERVATION KEY**	**APTPL**
REGISTER	ignored	ignored	valid	valid	valid
REGISTER AND IGNORE EXISTING KEY	ignored	ignored	ignored	valid	valid
RESERVE	LU_SCOPE	valid	valid	ignored	ignored
RELEASE	LU_SCOPE	valid	valid	ignored	ignored
CLEAR	ignored	ignored	valid	ignored	ignored
PREEMPT	LU_SCOPE	valid	valid	valid	ignored
PREEMPT AND ABORT	LU_SCOPE	valid	valid	valid	ignored
REGISTER AND MOVE	LU_SCOPE	valid	valid	valid	not applicable [a]

[a] The parameter list format for the REGISTER AND MOVE service action is described in 3.16.4.

Table 86. PERSISTENT RESERVE OUT service actions and valid parameters (part 2 of 2)

Service action	Allowed SCOPE	Parameters (part 2 of 2)	
		ALL_TG_PT	SPEC_I_PT
REGISTER	ignored	valid	valid
REGISTER AND IGNORE EXISTING KEY	ignored	valid	valid
RESERVE	LU_SCOPE	ignored	ignored
RELEASE	LU_SCOPE	ignored	ignored
CLEAR	ignored	ignored	ignored
PREEMPT	LU_SCOPE	ignored	ignored
PREEMPT AND ABORT	LU_SCOPE	ignored	ignored
REGISTER AND MOVE	LU_SCOPE	not applicable [a]	not applicable [a]

[a] The parameter list format for the REGISTER AND MOVE service action is described in 3.16.4.

3.16.4 PERSISTENT RESERVE OUT command with REGISTER AND MOVE service

The parameter list format shown in table 87 shall be used by the PERSISTENT RESERVE OUT command with REGISTER AND MOVE service action.

RESERVATION KEY field

The RESERVATION KEY field contains an 8-byte value provided by the application client to the device server to identify the I_T nexus that is the source of the PERSISTENT RESERVE OUT command. The device server shall verify that the contents of the RESERVATION KEY field in a PERSISTENT RESERVE OUT command parameter data matches the registered reservation key for the I_T nexus from which the command was received. If a PERSISTENT RESERVE OUT command specifies a RESERVATION KEY field other than the reservation key registered for the I_T nexus, the device server shall return a RESERVATION CONFLICT status.

SERVICE ACTION RESERVATION KEY field

The SERVICE ACTION RESERVATION KEY field contains the reservation key to be registered to the specified I_T nexus.

APTPL (Activate Persist Through Power Loss) bit

1 The Activate Persist Through Power Loss (APTPL) bit set to one is optional. If a device server that does not support an APTPL bit set to one receives that value, it shall return CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. If the last valid APTPL bit value received by the device server is one, the logical unit shall retain any persistent reservation(s) that may be present and all reservation keys (i.e., registrations) for all I_T nexuses even if power is lost and later returned (see SPC-4).

0 If the last valid APTPL bit value received by the device server is zero, the loss of power in the SCSI target device shall release the persistent reservation for the logical unit and remove all registered reservation keys (see SPC-4).

UNREG (Unregister) bit

0 The unregister (UNREG) bit set to zero specifies that the device server shall not unregister the I_T nexus on which the PERSISTENT RESERVE OUT command REGISTER AND MOVE service action was received.

1 An UNREG bit set to one specifies that the device server shall unregister the I_T nexus on which the PERSISTENT

Table 87. PERSISTENT RESERVE OUT command with REGISTER AND MOVE service

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)							
7	RESERVATION KEY							(LSB)
8	(MSB)							
15	SERVICE ACTION RESERVATION KEY							(LSB)
16	Reserved							
17	Reserved						UNREG	APTPL
18	(MSB)							
19	RELATIVE TARGET PORT IDENTIFIER							(LSB)
20	(MSB)							
23	TRANSPORTID PARAMETER DATA LENGTH (N - 23)							(LSB)
24								
n	TRANSPORTID							

RESERVE OUT command REGISTER AND MOVE service action was received.

RELATIVE TARGET PORT IDENTIFIER field

The RELATIVE TARGET PORT IDENTIFIER field specifies the relative port identifier of the target port in the I_T nexus to which the persistent reservation is to be moved.

TRANSPORTID DESCRIPTOR LENGTH field

The TRANSPORTID DESCRIPTOR LENGTH field specifies the number of bytes of the TRANSPORTID that follows, shall be a minimum of 24 bytes, and shall be a multiple of 4.

TRANSPORTID field

The TRANSPORTID field specifies the initiator port in the I_T nexus to which the persistent reservation is to be moved. The format of the TRANSPORTID is defined in SPC-4.

The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST:

a) If the value in the PARAMETER LIST LENGTH field in the CDB does not include all of the parameter list bytes specified by the TRANSPORTID PARAMETER DATA LENGTH field; or

b) If the value in the TRANSPORTID PARAMETER DATA LENGTH field results in the truncation of a TRANSPORTID.

3.17 PRE-FETCH (10) command, PRE-FETCH (16) command

These commands **are not** supported by Seagate disc drives. The command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID COMMAND OPERATION CODE.

3.18 PREVENT ALLOW MEDIUM REMOVAL command

This command **is not** supported by Seagate Disc drives. The drive returns CHECK CONDITION status with ILLEGAL REQUEST in the Sense Key.

The PREVENT ALLOW MEDIUM REMOVAL command (see SPC-4) requests that the logical unit enable or disable the removal of the medium. The logical unit shall not allow medium removal if any initiator port currently has medium removal prevented.

3.19 READ (6) command

The READ (6) command (see table 88) requests that the device server read the specified logical block(s) and transfer them to the data-in buffer. Each logical block read includes user data and, if the medium is formatted with protection information enabled, protection information. Each logical block transferred includes user data but does not include protection information. The most recent data value written, or to be written if cached, in the addressed logical blocks shall be returned.

Table 88. READ (6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (08h)							
1	Reserved			(MSB)				
2	LOGICAL BLOCK ADDRESS							
3								(LSB)
4	TRANSFER LENGTH							
5	CONTROL							

The cache control bits are not provided for this command. Direct-access block devices with cache may have values for the cache control bits that affect the READ (6) command; however, no default values are defined by this manual. If explicit control is required, the READ (10) command should be used.

LOGICAL BLOCK ADDRESS field

The LOGICAL BLOCK ADDRESS field specifies the first logical block accessed by this command. If the logical block address exceeds the capacity of the medium the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

TRANSFER LENGTH field

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be read and transferred to the data-in buffer, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A TRANSFER LENGTH field set to zero specifies that 256 logical blocks shall be read. Any other value specifies the number of logical blocks that shall be read. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page.

a) For the READ (10) command, READ (12) command, READ (16) command, and READ (32) command, a TRANSFER LENGTH field set to zero specifies that no logical blocks are read.

b) Although the READ (6) command is limited to addressing logical blocks up to a capacity of 2 Gigabytes, for block lengths of 512 bytes, this command has been maintained as mandatory since some system initialization routines require that the READ (6) command be used. System initialization routines should migrate from the READ (6) command to the READ (10) command, which is capable of addressing 2 Terabytes with block lengths of 512 bytes, or the READ (16) command to address more than 2 Terabytes.

The device server shall check the protection information read from the medium before returning status for the command as described in table 89.

Table 89. Protection information checking for READ (6)

Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [f]	Extended INQUIRY Data VPD page bit value [d]	If check fails [b] [c], additional sense code
Yes	No	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [a]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 [g]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
No		No protection information available to check		

[a] The device server checks the logical block application tag only if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. The method for acquiring this knowledge is not defined by this manual.
[b] If an error is reported, the sense key shall be set to ABORTED COMMAND.
[c] If multiple errors occur, the selection of which error to report is not defined by this manual.
[d] See the Extended INQUIRY Data VPD page (see SPC-4) for the definitions of the GRD_CHK bit, APP_CHK bit, and REF_CHK bit.
[e] If the device server detects a:

a) LOGICAL BLOCK APPLICATION TAG field set to FFFFh and type 1 protection (See SBC-3) or type 2 protection (See SBC-3) is enabled; or
b) LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFF FFFFh, and type 3 protection (See SBC-3) is enabled, then the device server shall not check any protection information in the associated logical block.

[f] If type 1 protection is enabled, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the logical block reference tag only if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field. The method for acquiring this knowledge is not defined by this manual.

3.20 READ (10) command

The READ (10) command (see table 90) requests that the device server read the specified logical block(s) and transfer them to the data-in buffer. Each logical block read includes user data and, if the medium is formatted with protection information enabled, protection information. Each logical block transferred includes user data and may include protection information, based on the RDPROTECT field and the medium format. The most recent data value written in the addressed logical block shall be returned.

Table 90. READ (10) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (28h)							
1	RDPROTECT			DPO	FUA	Reserved	FUA_NV	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved			GROUP NUMBER				
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

RDPROTECT field

The device server shall check the protection information read from the medium before returning status for the command based on the RDPROTECT field as described in table 91.

Table 91. RDPROTECT field (Sheet 1 of 3)

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extended INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
000b	Yes	No	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
				GRD_CHK = 0	NO CHECK PERFORMED
			LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
				APP_CHK = 0	NO CHECK PERFORMED
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 [i]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
				REF_CHK = 0	NO CHECK PERFORMED
	No		No protection information available to check		
001b 101b [b]	Yes	Yes [e]	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
				GRD_CHK = 0	NO CHECK PERFORMED
			LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
				APP_CHK = 0	NO CHECK PERFORMED
			LOGICAL BLOCK REFERENCE TAG [l]	REF_CHK = 1 [i]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
				REF_CHK = 0	NO CHECK PERFORMED
	No [a]	No protection information available to transmit to the data-in buffer or for checking			

Table 91. RDPROTECT field (Sheet 2 of 3)

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extended INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
010b [b]	Yes	Yes [e]	LOGICAL BLOCK GUARD	NO CHECK PERFORMED	
			LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
				APP_CHK = 0	NO CHECK PERFORMED
			LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 [i]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
				REF_CHK = 0	NO CHECK PERFORMED
	No [a]	No protection information available to transmit to the data-in buffer or for checking			
011b [b]	Yes	Yes [e]	LOGICAL BLOCK GUARD	NO CHECK PERFORMED	
			LOGICAL BLOCK APPLICATION TAG	NO CHECK PERFORMED	
			LOGICAL BLOCK REFERENCE TAG	NO CHECK PERFORMED	
	No [a]	No protection information available to transmit to the data-in buffer or for checking			

Table 91. RDPROTECT field (Sheet 3 of 3)

Code	Logical unit formatted with protection information	Shall device server transmit protection information?	Field in protection information [h]	Extended INQUIRY Data VPD page bit value [g]	If check fails [d] [f], additional sense code
100b [b]	Yes	Yes [e]	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
				GRD_CHK = 0	NO CHECK PERFORMED
			LOGICAL BLOCK APPLICATION TAG	NO CHECK PERFORMED	
			LOGICAL BLOCK REFERENCE TAG	NO CHECK PERFORMED	
	No [a]	No protection information available to transmit to the data-in buffer or for checking			
101b - 111b	Reserved				

[a] A read operation to a logical unit that supports protection information and has not been formatted with protection information 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.

[b] If the logical unit does not support protection information the requested command should 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.

[c] The device server shall check the logical block application tag if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the READ (32) command (see 3.23) is used and the ATO bit is set to one in the Control mode page (see SPC-4), this knowledge is acquired from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge may be acquired by a method not defined by this manual.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] Transmit protection information to the data-in buffer.

[f] If multiple errors occur, the selection of which error to report is not defined by this manual.

[g] See the Extended INQUIRY Data VPD page (see SPC-4) for the definitions of the GRD_CHK bit, the APP_CHK bit, and the REF_CHK bit.

[h] If the device server detects:

a) a LOGICAL BLOCK APPLICATION TAG field set to FFFFh and type 1 protection (See SBC-3) or type 2 protection (See SBC-3) is enabled; or

1) a LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFF FFFFh, and type 3 protection (See SBC-3) is enabled,

then the device server shall not check any protection information in the associated protection information interval.

i If type 1 protection is enabled, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the LOGICAL BLOCK REFERENCE TAG if it has knowledge of the contents of the logical block reference tag field. If type 2 protection is enabled, then this knowledge may be acquired through the expected INITIAL LOGICAL BLOCK REFERENCE TAG field in a READ (32) command (see 3.23). If type 3 protection is enabled, then the method for acquiring this knowledge is not defined by this manual.

DPO (Disable Page Out) bit

0 A Disable Page Out (DPO) bit set to zero specifies that the retention priority shall be determined by the RETENTION PRIORITY fields in the Caching mode page (see 4.3.7).

1 A DPO bit set to one specifies that the device server shall assign the logical blocks accessed by this command the lowest retention priority for being fetched into or retained by the cache. A DPO bit set to one overrides any retention priority specified in the Caching mode page. All other aspects of the algorithm implementing the cache replacement strategy are not defined by this manual.

Note. The DPO bit is used to control replacement of logical blocks in the cache when the application client has information on the future usage of the logical blocks. If the DPO bit is set to one, the application client is specifying that the logical blocks accessed by the command are not likely to be accessed again in the near future and should not be put in the cache nor retained by the cache. If the DPO bit is set to zero, the application client is specifying that the logical blocks accessed by this command are likely to be accessed again in the near future.

The force unit access (FUA) and force unit access non-volatile cache (FUA_NV) bits are defined in table 92.

Table 92. Force unit access for read operations

FUA	FUA_NV	Description
0	0	The device server may read the logical blocks from volatile cache, non-volatile cache, and/or the medium.
0	1	If the NV_SUP bit is set to one in the Extended INQUIRY Data VPD page (see 4.4.6), the device server shall read the logical blocks from non-volatile cache or the medium. If a non-volatile cache is present and a volatile cache contains a more recent version of a logical block, the device server shall write the logical block to: (a) non-volatile cache; and/or (b) the medium, before reading it. If the NV_SUP bit is set to zero in the Extended INQUIRY Data VPD page (see 4.4.6), the device server may read the logical blocks from volatile cache, non-volatile cache, and/or the medium.
1	0 or 1	The device server shall read the logical blocks from the medium. If a cache contains a more recent version of a logical block, the device server shall write the logical block to the medium before reading it.

LOGICAL BLOCK ADDRESS field

The LOGICAL BLOCK ADDRESS field specifies the first logical block accessed by this command. If the logical block address exceeds the capacity of the medium the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

GROUP NUMBER field

The GROUP NUMBER field specifies the group into which attributes associated with the command should be collected. A GROUP NUMBER field set to zero specifies that any attributes associated with the command shall not be collected into any group.

TRANSFER LENGTH field

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be read and transferred to the data-in buffer, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A TRANSFER LENGTH field set to zero specifies that no logical blocks shall be read. This condition shall not be considered an error. Any other value specifies the number of logical blocks that shall be read. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page.

3.21 READ (12) command

The READ (12) command (see table 93) requests that the device server read the specified logical block(s) and transfer them to the data-in buffer. Each logical block read includes user data and, if the medium is formatted with protection information enabled, protection information. Each logical block transferred includes user data and may include protection information, based on the RDPROTECT field and the medium format.

Table 93. READ (12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A8h)							
1	RDPROTECT			DPO	FUA	Reserved	FUA_NV	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	RESTRICTED FOR MMC-6	Reserved		GROUP NUMBER				
11	CONTROL							

See the READ (10) command for the definitions of the fields in this command.

3.22 READ (16) command

The READ (16) command (see table 94) requests that the device server read the specified logical block(s) and transfer them to the data-in buffer. Each logical block read includes user data and, if the medium is formatted with protection information enabled, protection information. Each logical block transferred includes user data and may include protection information, based on the RDPROTECT field and the medium format.

Table 94. READ (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (88h)							
1	RDPROTECT			DPO	FUA	Reserved	FUA_NV	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	TRANSFER LENGTH						
13								(LSB)
14	Restricted for MMC-6	Reserved		GROUP NUMBER				
15	CONTROL							

See the READ (10) command for the definitions of the fields in this command.

3.23 READ (32) command

The READ (32) command (see table 95) requests that the device server read the specified logical block(s) and transfer them to the data-in buffer. Each logical block read includes user data and, if the medium is formatted with protection information enabled, protection information. Each logical block transferred includes user data and may include protection information, based on the RDPROTECT field and the medium format.

1. The READ (32) command shall only be processed if type 2 protection is enabled (see SBC-3).

Table 95. READ (32) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2 – 5	Reserved							
6	Reserved			GROUP NUMBER				
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (0009h)						
9								(LSB)
10	RDPROTECT			DPO	FUA	Reserved	FUA_NV	Reserved
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23								(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK						
27								(LSB)
28	(MSB)	TRANSFER LENGTH						
31								(LSB)

See the READ (10) command for the definitions of the GROUP NUMBER field, the RDPROTECT field, the DPO bit, the FUA bit, the FUA_NV bit, the LOGICAL BLOCK ADDRESS field, and the TRANSFER LENGTH field.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 91), the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA.

ATO bit

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 91), the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK field bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in the protection information.

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see table 91), or if the ATO bit is set to zero, the LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored.

3.24 READ BUFFER command

3.24.1 READ BUFFER command introduction

The READ BUFFER command (see table 96) is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing memory in the SCSI device and the integrity of the service delivery subsystem. This command shall not alter the medium.

Table 96. READ BUFFER command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Ch)							
1	Reserved			MODE				
2	BUFFER ID							
3	(MSB)							
	BUFFER OFFSET							
5								(LSB)
6	(MSB)							
	ALLOCATION LENGTH							
8								(LSB)
9	CONTROL							

MODE field

The function of this command and the meaning of fields within the CDB depend on the contents of the MODE field. The MODE field is defined in table 97.

Table 97. MODE field

MODE	Description
00h	Combined header and data [a]
01h	Not supported [a]
02h	Data
03h	Descriptor
04h - 09h	Reserved
0Ah	Echo buffer
0Bh	Echo buffer descriptor
0Ch - 19h	Reserved
1Ah	Enable expander communications protocol and Echo buffer
1Bh	Reserved
1Ch	Error history
1Dh - 1Fh	Reserved

[a] Mode 00h and 01h are not recommended.

If the mode is not set to one, the ALLOCATION LENGTH field is defined in 2.2.6.

The four-byte READ BUFFER header (see table 98) is followed by data bytes from the buffer.

Table 98. READ BUFFER header

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	(MSB)	BUFFER CAPACITY						
3								(LSB)
4	DATA							
n								

BUFFER CAPACITY field

The BUFFER CAPACITY field specifies the total number of data bytes available in the buffer. The buffer capacity is not reduced to reflect the actual number of bytes written using the WRITE BUFFER command with combined header and data mode. The relationship between the BUFFER CAPACITY field and the CDB ALLOCATION LENGTH field is defined in 2.2.6. Following the READ BUFFER header, the device server shall transfer data from the buffer.

3.24.1.1 Vendor specific mode (01h)

This mode is not supported by Seagate products. If used, the drive will return CHECK CONDITION and report an ILLEGAL REQUEST/INVALID FIELD IN CDB error.

3.24.1.2 Data mode (02h)

In this mode, the Data-In Buffer is filled only with logical unit buffer data. The BUFFER ID field specifies a buffer within the logical unit from which data shall be transferred. Seagate assigns buffer ID codes to buffers within the logical unit. Buffer ID zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. Buffer ID code assignments for the READ BUFFER command shall be the same as for the WRITE BUFFER command. If an unsupported buffer ID code is selected, then the 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.

The BUFFER OFFSET field contains the byte offset within the specified buffer from which data shall be transferred. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor (see 3.24.1.3). If the device server is unable to accept the specified buffer offset, the 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.

3.24.1.3 Descriptor mode (03h)

In this mode, a maximum of four bytes of READ BUFFER descriptor information is returned. The device server shall return the descriptor information for the buffer specified by the BUFFER ID field (see the description of the buffer ID in 3.24.1.2). If there is no buffer associated with the specified buffer ID, the device server shall return all zeros in the READ BUFFER descriptor. The BUFFER OFFSET field is reserved in this mode. The allocation length should be set to four or greater. The READ BUFFER descriptor is defined as shown in table 99.

Table 99. READ BUFFER descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	OFFSET BOUNDARY							
1	(MSB)							
3	BUFFER CAPACITY							(LSB)

OFFSET BOUNDARY field

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

For READ BUFFER commands, the OFFSET BOUNDARY field (see table 100) applies to the following modes:

a) data (i.e., 02h) (see 3.24.1.2); and
b) error history (i.e., 1Ch) (see 3.24.1.7)

BUFFER OFFSET field

The value contained in the BUFFER OFFSET field of subsequent WRITE BUFFER and READ BUFFER commands should be a multiple of $2^{\text{offset boundary}}$ as shown in table 100.

Table 100. Buffer offset boundary

Offset boundary	$2^{\text{Offset boundary}}$	Buffer offsets
0h	$2^0 = 1$	Byte boundaries
1h	$2^1 = 2$	Even-byte boundaries
2h	$2^2 = 4$	Four-byte boundaries
3h	$2^3 = 8$	Eight-byte boundaries
4h	$2^4 = 16$	16-byte boundaries
.	.	.
FFh	Not applicable	0 is the only supported buffer offset

BUFFER CAPACITY field

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

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

3.24.1.4 Echo buffer mode (0Ah)

In this mode the device server transfers data to the application client from the echo buffer that was written by the most recent WRITE BUFFER command with the mode field set to echo buffer received on the same I_T nexus. The READ BUFFER command shall return the same number of bytes of data as received in the prior WRITE BUFFER command with the mode field set to echo buffer, limited by the allocation length. The BUFFER ID and BUFFER OFFSET fields are ignored in this mode.

If no WRITE BUFFER command with the mode set to echo buffer received on this I_T nexus has completed without an error, then the READ BUFFER command shall terminate with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR. If the data in the echo buffer has been overwritten by another I_T nexus, the READ BUFFER command shall be terminated with CHECK CONDITION status, with the sense key set to ABORTED COMMAND, and the additional sense code set to ECHO BUFFER OVERWRITTEN.

After a WRITE BUFFER command with the mode set to echo buffer has completed without an error, the application client may send multiple READ BUFFER commands with the mode set to echo buffer in order to read the echo buffer data multiple times.

3.24.1.5 Echo buffer descriptor mode (0Bh)

In this mode, a maximum of four bytes of READ BUFFER descriptor information is returned. The device server shall return the descriptor information for the echo buffer. If there is no echo buffer implemented, the device server shall return all zeros in the READ BUFFER descriptor. The BUFFER ID field and BUFFER OFFSET field are reserved in this mode. The allocation length should be set to four or greater. The READ BUFFER descriptor is defined as shown in table 101.

Table 101. Echo buffer descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							EBOS
1	Reserved							
2	Reserved			(MSB)				
3	BUFFER CAPACITY							(LSB)

BUFFER CAPACITY field

The BUFFER CAPACITY field shall return the size of the echo buffer in bytes aligned to a four-byte boundary. The maximum echo buffer size is 4 096 bytes.

If the echo buffer is implemented, the echo buffer descriptor shall be implemented.

EBOS bit

1 An echo buffer overwritten supported (EBOS) bit set to one indicates either:

a) The device server returns the ECHO BUFFER OVERWRITTEN additional sense code if the data being read from the echo buffer is not the data previously written by the same I_T nexus, or

b) The device server ensures echo buffer data returned to each I_T nexus is the same as that previously written by that I_T nexus.

0 An EBOS bit set to zero specifies that the echo buffer may be overwritten by any intervening command received on any I_T nexus.

A READ BUFFER command with the mode set to echo buffer descriptor may be used to determine the echo buffer capacity and supported features before a WRITE BUFFER command with the mode set to echo buffer is sent.

3.24.1.6 Enable expander communications protocol and Echo buffer (1Ah)

Receipt of a READ BUFFER command with this mode (1Ah) causes a communicative expander (see SPI-5) to enter the expanded communications protocol mode. Device servers in SCSI target devices that receive a READ BUFFER command with this mode shall process it as if it were a READ BUFFER command with mode 0Ah (see 3.24.1.4).

3.24.1.7 Error history mode (1Ch)

3.24.1.7.1 Error history overview

This mode is used to manage and retrieve error history (see SPC-4).

If the device server is unable to process a READ BUFFER command with the MODE field set to 1Ch because of a vendor specific condition, then the device server shall terminate the READ BUFFER command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

The BUFFER ID field (see table 102) specifies the action that the device server shall perform, and the parameter data, if any, that the device server shall return.

Table 102. Error history BUFFER ID field

Code	Description	Buffer offset	Error history I_T nexus constrained	Reference
00h	Return error history directory	0000h	Yes	3.24.1.7.2
01h	Return error history directory and create new error history snapshot (see 1.1and SPC-4)	0000h	Yes	3.24.1.7.2
02h	Return error history directory and establish new error history I_T nexus (see 1.1and SPC-4)	0000h	No	3.24.1.7.2
03h	Return error history directory, establish new error history I_T nexus, and create new error history snapshot	0000h	No	3.24.1.7.2
04h to 0Fh	Reserved		Yes	
10h to EFh	Return error history	0000h to FFFFh	Yes	3.24.1.7.3
F0h to FDh	Reserved		Yes	
FEh	Clear error history I_T nexus	Ignored	Yes	3.24.1.7.4
FFh	Clear error history I_T nexus and release error history snapshot	Ignored	Yes	3.24.1.7.5

The command shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to OPERATION IN PROGRESS if the device server receives a READ BUFFER command:

a) with the MODE field set to 1Ch;
b) with the BUFFER ID field set to a value that table 102 shows as constrained by error history I_T nexus;
c) if an error history I_T nexus exists and the command is received from an I_T nexus that is different that I_T nexus; and
d) an error history snapshot exists.

BUFFER OFFSET field

The BUFFER OFFSET field specifies the byte offset from the start of the buffer specified by the BUFFER ID field from which the device server shall return data. The application client should conform to the offset boundary requirements indicated in the READ BUFFER descriptor (see 3.24.1.3). If the buffer offset is not one of those shown in table 102 or the device server is unable to accept the specified buffer offset, then the device server shall terminate the READ BUFFER command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

3.24.1.7.2 Error history directory

Whenever allowed by established error history I_T nexus constraints (see 3.24.1.7.1), if any, all error history device server actions return an error history directory (see table 104). Some error history device server actions also discard the existing error history snapshot (see SPC-4) and create a new error history snapshot (see table 103).

Table 103 — Summary of error history directory device server actions

BUFFER ID field	Establish new error history I_T nexus (see SPC-4)	Error history snapshot (see SPC-4)	
		Preserved (if exists)	Created
00h	No [a]	Yes	No [b]
01h	No [a]	No	Yes
02h	Yes	Yes	No [b]
03h	Yes	No	Yes

[a] If no error history I_T nexus is established, a new one is established.
[b] If no error history snapshot exists, a new one is created.

The error history directory is defined in table 104.

Table 104. Error history directory

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	T10 VENDOR INDENTIFICATION						
7								(LSB)
8	VERSION							
9	Reserved			EHS_RETRIEVED		EHS_SOURCE		CLR_SUP
10	Reserved							
29								
30	(MSB)	DIRECTORY LENGTH (n-31)						
31								(LSB)
	Error history directory list							
32	Error history directory entry [first] (see table 107)							
39								
	. . .							
n-7	Error history directory entry [last] (see table 107)							
n								

T10 VENDOR INDENTIFICATION field

The T10 VENDOR INDENTIFICATION field contains eight bytes of left-aligned ASCII data identifying the manufacturer of the logical unit. The T10 vendor identification shall be one assigned by INCITS.

Note. The T10 VENDOR INDENTIFICATION field may contain a different value than the VENDOR INDENTIFICATION field in the standard INQUIRY data (see 3.6.2) (e.g., this field may indicate a disk drive component vendor while the standard INQUIRY data indicates the original equipment manufacturer).

VERSION field

The VERSION field indicates the version and format of the vendor specific error history. The VERSION field is assigned by the vendor indicated in the T10 VENDOR INDENTIFICATION field.

EHS_RETRIEVED (error history retrieved) field

The error history retrieved (EHS_RETRIEVED) field (see table 105) indicates whether a clear error history device server action has been requested for the error history snapshot. EHS_RETRIEVED field shall be set to 00b or 10b when the error history snapshot is created.

Table 105. EHS_RETRIEVED field

Code	Description
00b	No information
01b	The error history I_T nexus has requested buffer ID FEh (i.e., clear error history I_T nexus) or buffer ID FFh (i.e., clear error history I_T nexus and release snapshot) for the current error history snapshot.
10b	An error history I_T nexus has not requested buffer ID FEh (i.e., clear error history I_T nexus) or buffer ID FFh (i.e., clear error history I_T nexus and release snapshot) for the current error history snapshot.
11b	Reserved

EHS_SOURCE (error history source) field

The error history source (EHS_SOURCE) field (see table 106) indicates the source of the error history snapshot.

Table 106 — EHS_SOURCE field

Code	Description
00b	The error history snapshot was created by the device server and was not created due to processing a READ BUFFER command.
01b	Error history snapshot was created due to processing of the current READ BUFFER command
10b	Error history snapshot was created due to processing of a previous READ BUFFER command
11b	Reserved

CLR_SUPC (clear support) bit

1 A clear support (CLR_SUP) bit set to one indicates that the CLR bit is supported in the WRITE BUFFER command download error history mode (see 3.70).

0 A CLR_SUP bit set to zero indicates that the CLR bit is not supported.

DIRECTORY LENGTH field

The DIRECTORY LENGTH field indicates the number of error history directory list bytes available to be transferred. This value shall not be altered even if the allocation length is not sufficient to transfer the entire error history directory list.

The error history directory list contains an error history directory entry (see table 107) for each supported buffer ID in the range of 00h to EFh. The first entry shall be for buffer ID 00h and the entries shall be in order of ascending buffer IDs. The supported buffer IDs are not required to be contiguous. There shall not be any entries for buffer IDs greater than or equal to F0h.

Table 107. Error history directory entry

Bit / Byte	7	6	5	4	3	2	1	0
0	SUPPORTED BUFFER ID							
1	Reserved							
3								
4	(MSB)	MAXIMUM AVAILABLE LENGTH						
7								(LSB)

SUPPORTED BUFFER ID field

The SUPPORTED BUFFER ID field indicates the error history buffer ID associated with this entry.

MAXIMUM AVAILABLE LENGTH

The MAXIMUM AVAILABLE LENGTH field indicates the maximum number of data bytes contained in the buffer indicated by the SUPPORTED BUFFER ID field. The actual number of bytes available for transfer may be smaller.

3.24.1.7.3 Error history data buffer

Unless an error is encountered, the device server shall return parameter data that contains error history in a vendor specific format from the error history snapshot from the specified buffer at the specified buffer offset.

If the device server receives a READ BUFFER command with the MODE field set to 1Ch from the established error history I_T nexus and the BUFFER ID field is set to a value that the error history directory (see 3.24.1.7.2) shows as not supported, then the 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.

If the value in the BUFFER OFFSET field is not supported, the 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.

The amount of error history in the specified buffer shall be less than or equal to the number of bytes indicated by the MAXIMUM AVAILABLE LENGTH field in the error history directory (see 3.24.1.7.2).

3.24.1.7.4 Clear error history I_T nexus

If the BUFFER ID field is set to FEh, the device server shall:

a) clear the error history I_T nexus, if any; and
b) not transfer any data.

3.24.1.7.5 Clear error history I_T nexus and release snapshot

If the BUFFER ID field is set to FFh, the device server shall:

a) clear the error history I_T nexus, if any,
b) release the error history snapshot, if any; and
c) not transfer any data.

3.25 READ CAPACITY (10) command

3.25.1 READ CAPACITY (10) overview

The READ CAPACITY (10) command (see table 108) requests that the device server transfer 8 bytes of parameter data describing the capacity and medium format of the direct-access block device to the data-in buffer. This command may be processed as if it has a HEAD OF QUEUE task attribute. If the logical unit supports protection information, the application client should use the READ CAPACITY (16) command instead of the READ CAPACITY (10) command.

Table 108. READ CAPACITY (10) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (25h)							
1	Reserved							Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved							
7								
8	Reserved							PMI
9	CONTROL							

For the definition of the LOGICAL BLOCK ADDRESS field see 2.2.3.

LOGICAL BLOCK ADDRESS field

The LOGICAL BLOCK ADDRESS field shall be set to zero if the PMI bit is set to zero. If the PMI bit is set to zero and the LOGICAL BLOCK ADDRESS field is not set to zero, the device server 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 (Partial Medium Indicator) bit

0 A partial medium indicator (PMI) bit set to zero specifies that the device server return information on the last logical block on the direct-access block device.

1 A PMI bit set to one specifies that the device server return information on the last logical block after that specified in the LOGICAL BLOCK ADDRESS field before a substantial vendor-specific delay in data transfer may be encountered.

This function is intended to assist storage management software in determining whether there is sufficient space starting with the logical block address specified in the CDB to contain a frequently accessed data structure (e.g., a file directory or file index) without incurring an extra delay.

3.25.2 READ CAPACITY (10) parameter data

The READ CAPACITY (10) parameter data is defined in table 109. Any time the READ CAPACITY (10) parameter data changes, the device server should establish a unit attention condition as described in SBC-3.

Table 109. READ CAPACITY (10) parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				RETURNED LOGICAL BLOCK ADDRESS				
3								(LSB)
4	(MSB)							
				BLOCK LENGTH IN BYTES				
7								(LSB)

RETURNED LOGICAL BLOCK ADDRESS field

If the number of logical blocks exceeds the maximum value that is able to be specified in the RETURNED LOGICAL BLOCK ADDRESS field, the device server shall set the RETURNED LOGICAL BLOCK ADDRESS field to FFFFFFFFh. The application client should then issue a READ CAPACITY (16) command (see 3.27) to retrieve the READ CAPACITY (16) parameter data.

0 If the PMI bit is set to zero, the device server shall set the RETURNED LOGICAL BLOCK ADDRESS field to the lower of:

a) the LBA of the last logical block on the direct-access block device; or

b) FFFFFFFFh.

1 If the PMI bit is set to one, the device server shall set the RETURNED LOGICAL BLOCK ADDRESS field to the lower of:

a) the last LBA after that specified in the LOGICAL BLOCK ADDRESS field of the CDB before a substantial vendor-specific delay in data transfer may be encountered; or

b) FFFFFFFFh.

The RETURNED LOGICAL BLOCK ADDRESS shall be greater than or equal to that specified by the LOGICAL BLOCK ADDRESS field in the CDB.

BLOCK LENGTH IN BYTES field

The BLOCK LENGTH IN BYTES field contains the number of bytes of user data in the logical block indicated by the RETURNED LOGICAL BLOCK ADDRESS field. This value does not include protection information or additional information (e.g., ECC bytes) recorded on the medium.

3.26 READ CAPACITY (16) command

3.26.1 READ CAPACITY (16) command overview

The READ CAPACITY (16) command (see table 110) requests that the device server transfer parameter data describing the capacity and medium format of the direct-access block device to the data-in buffer. This command is mandatory if the logical unit supports protection information and optional otherwise (see SBC-3). This command is implemented as a service action of the SERVICE ACTION IN operation code. This command may be processed as if it has a HEAD OF QUEUE task attribute.

Table 110. READ CAPACITY (16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Eh)							
1	Reserved			SERVICE ACTION (10h)				
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	ALLOCATION LENGTH						
13								(LSB)
14	Reserved							PMI
15	CONTROL							

LOGICAL BLOCK ADDRESS field

See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field.

See the READ CAPACITY (10) command (see 3.25) for the definition of the PMI bit.

ALLOCATION LENGTH field

The ALLOCATION LENGTH field (see 2.2.6) specifies the maximum number of bytes that the application client has allocated for returned parameter data. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error. The device server shall terminate transfers to the data-in buffer when the number of bytes specified by the ALLOCATION LENGTH field have been transferred or when all available data has been transferred, whichever is less. The contents of the parameter data shall not be altered to reflect the truncation, if any, that results from an insufficient allocation length.

3.26.2 READ CAPACITY (16) parameter data

The READ CAPACITY (16) parameter data is defined in table 111. Any time the READ CAPACITY (16) parameter data changes, the device server should establish a unit attention condition as described in SPC-4.

Table 111. READ CAPACITY (16) parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	RETURNED LOGICAL BLOCK ADDRESS						
7								(LSB)
8	(MSB)	LOGICAL BLOCK LENGTH IN BYTES						
11								(LSB)
12	Reserved				P_TYPE			PROT_EN
13	P_I_EXPONENT				LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT			
14	TPE	TPRZ	(MSB)	LOWEST ALIGNED LOGICAL BLOCK ADDRESS				
15								(LSB)
16	Reserved							
31								

RETURNED LOGICAL BLOCK ADDRESS

The RETURNED LOGICAL BLOCK ADDRESS field and LOGICAL BLOCK LENGTH IN BYTES field of the READ CAPACITY (16) parameter data are the same as the in the READ CAPACITY (10) parameter data (see 3.25.1). The maximum value that shall be returned in the RETURNED LOGICAL BLOCK ADDRESS field is FFFFFFFF_FFFFFFFEh.

P_TYPE (Protection Type) field

The protection type (P_TYPE) field and the PROT_EN bit (see table 112) indicate the logical unit's current type of protection.

Table 112. P_TYPE FIELD and PROT_EN bit

PROT_EN	P_TYPE	DESCRIPTION
0	XXXb	The logical unit is formatted to type 0 protection (see SPC-4)
1	000b	The logical unit is formatted to type 1 protection (see SPC-4)
1	001b	The logical unit is formatted to type 2 protection (see SPC-4)
1	010b	The logical unit is formatted to type 3 protection (see SPC-4)
1	011b - 111b	Reserved

P_I_EXPONENT field

The P_I_EXPONENT field may be used to determine the number of protection information intervals placed within each logical block (see 3.5.2).

The number of protection information intervals is calculated as follows:

$$\text{number of protection information intervals} = 2^{(\text{p\_i exponent})}$$

where:

p_i exponent is the contents of the P_I EXPONENT field

LOGICAL BLOCKS PER PHYSICAL BLOCKS EXPONENT field

The LOGICAL BLOCKS PER PHYSICAL BLOCKS EXPONENT field is defined in table 113.

Table 113. LOGICAL BLOCKS PER PHYSICAL BLOCKS EXPONENT field

Code	Description
0	One or more physical blocks per logical block [a]
n > 0	2^n logical blocks per physical block

[a] The number of physical blocks per logical block is not reported.

Thin Provisioning Enabled (TPE) bit

1 If the thin provisioning enabled (TPE) bit is set to one, then the logical unit implements thin provisioning (see SBC-3).

0 If the TPE bit is set to zero, then the logical unit implements full provisioning (see SBC-3).

Thin Provisioning Read Zeros (TPRZ) bit

1 If the thin provisioning read zeros (TPRZ) bit is set to one, then, for an unmapped LBA specified by a read operation, the device server shall send user data with all bits set to zero to the data-in buffer.

0 If the TPRZ bit is set to zero, then, for an unmapped LBA specified by a read operation, the device server shall send user data with all bits set to any value to the data-in buffer.

LOWEST ALIGNED LOGICAL BLOCK ADDRESS field

The LOWEST ALIGNED LOGICAL BLOCK ADDRESS field indicates the LBA of the first logical block that is located at the beginning of a physical block (SBC-3).

Note. The highest LBA that the lowest aligned logical block address field supports is 3FFFh (i.e., 16 383).

3.27 READ DEFECT DATA (10) command

3.27.1 READ DEFECT DATA (10) command overview

The READ DEFECT DATA (10) command (see table 114) requests that the device server transfer the medium defect data to the data-in buffer.

Table 114. READ DEFECT DATA (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (37h)							
1	Reserved							
2	Reserved			REQ_PLIST	REQ_GLIST	DEFECT LIST FORMAT		
3	Reserved							
6								
7	(MSB)	ALLOCATION LENGTH						
8								(LSB)
9	CONTROL							

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

Note. Some device servers may not be able to return medium defect data until after a FORMAT UNIT command (see 3.5) has been completed successfully.

REQ_PLIST (request primary defect list) bit

0 A request primary defect list (REQ_PLIST) bit set to zero specifies that the device server shall not return the PLIST.

1 A REQ_PLIST bit set to one specifies that the device server shall return the PLIST, if any.

REQ_GLIST (request grown defect list) bit

0 A request grown defect list (REQ_GLIST) bit set to zero specifies that the device server shall not return the GLIST.

1 A REQ_GLIST bit set to one specifies that the device server shall return the GLIST, if any.

A REQ_PLIST bit set to zero and a REQ_GLIST bit set to zero specifies that the device server shall return only the defect list header (i.e., the first four bytes of the defect list).

A REQ_PLIST bit set to one and a REQ_GLIST bit set to one specifies that the device server shall return both the PLIST and GLIST, if any. The order the lists are returned in is vendor-specific. Whether the lists are merged or not is vendor-specific.

DEFECT LIST FORMAT field

The DEFECT LIST FORMAT field specifies the preferred format for the defect list. This field is intended for those device servers capable of returning more than one format, as defined in the FORMAT UNIT command (see 3.5.5). A device server unable to return the requested format shall return the defect list in its default format and indicate that format in the DEFECT LIST FORMAT field in the defect list header (see table 115).

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

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6. The application client is responsible for comparing the allocation length requested in the CDB with the defect list length returned in the parameter data to determine whether a partial list was received. If the number of address descriptors the device server has to report exceeds the maximum value that is able to be specified in the ALLOCATION LENGTH field, the device server shall transfer no data and return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

3.27.2 READ DEFECT DATA (10) parameter data

The READ DEFECT DATA (10) parameter data (see table 115) contains a four-byte header, followed by zero or more address descriptors.

Table 115. READ DEFECT DATA (10) parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved			PLISTV	GLISTV	DEFECT LIST FORMAT		
2	(MSB)	DEFECT LIST LENGTH (N - 3)						
3								(LSB)
Defect list (if any)								
4	ADDRESS DESCRIPTOR(S) (IF ANY)							
n								

PLISTV (PLIST valid) bit

0 A PLIST valid (PLISTV) bit set to zero indicates that the data returned does not contain the PLIST.

1 A PLISTV bit set to one indicates that the data returned contains the PLIST.

GLISTV (GLIST valid) bit

0 A GLIST valid (GLISTV) bit set to zero indicates that the data returned does not contain the GLIST.

1 A GLISTV bit set to one indicates that the data returned contains the GLIST.

DEFECT LIST FORMAT field

The DEFECT LIST FORMAT field indicates the format of the address descriptors returned by the device server. This field is defined in the FORMAT UNIT command (see clause 3.5).

If the device server returns short block format address descriptors or long block format address descriptors, the address descriptors contain vendor-specific values.

Note. The use of the short block format and the long block format is not recommended for this command. There is no standard model that defines the meaning of the block address of a defect. In the usual case, a defect that has been reassigned no longer has an LBA.

If the device server returns physical sector format address descriptors (see 3.5.5.5), it may or may not include defects in parts of the medium not accessible to the application client. If the device server returns bytes from index format address descriptors (see 3.5.5.4), it shall return a complete list of the defects. A complete list of the defects may include defects in areas not within the capacity returned in the READ CAPACITY command.

DEFECT LIST LENGTH field

The DEFECT LIST LENGTH field indicates the length in bytes of the address descriptors that follow. The DEFECT LIST LENGTH is equal to four or eight times the number of the address descriptors, depending on the format of the returned address descriptors.

The address descriptors may or may not be sent in ascending order.

3.28 READ DEFECT DATA (12) command

3.28.1 READ DEFECT DATA (12) command overview

The READ DEFECT DATA (12) command (see Table 116) requests that the device server transfer the medium defect data to the data-in buffer.

Table 116. READ DEFECT DATA (12) command

Byte \ Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (B7h)							
1	Reserved			REQ_PLIST	REQ_GLIST	DEFECT LIST FORMAT		
2	Reserved							
5								
6	(MSB)	ALLOCATION LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

See the READ DEFECT DATA (10) command (see 3.27) for the definitions of the fields in this command.

Note. The application client may determine the length of the defect list by sending the READ DEFECT DATA (12) command with an ALLOCATION LENGTH field set to eight. The device server returns the defect list header that contains the length of the defect list.

3.28.2 READ DEFECT DATA (12) parameter data

The READ DEFECT DATA (12) parameter data (see table 117) contains an eight byte header, followed by zero or more address descriptors.

Table 117. READ DEFECT DATA (12) parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Reserved			PLISTV	GLISTV	DEFECT LIST FORMAT		
2	Reserved							
3	Reserved							
4	(MSB)	DEFECT LIST LENGTH (N - 7)						
7								(LSB)
Defect list (if any)								
8	ADDRESS DESCRIPTOR(S) (IF ANY)							
n								

See the READ DEFECT DATA (10) command (see 3.27) for the definitions of the fields in the defect list.

3.29 READ LONG (10) command

The READ LONG (10) command (see table 118) requests that the device server transfer data from a single logical block to the data-in buffer. The data transferred during the READ LONG (10) command is vendor-specific, but shall include the following items recorded on the medium:

a) if a logical block is being transferred, then:
   A) user data or transformed user data for the logical block;
   B) protection information or transformed protection information, if any, for the logical block; and
   C) any additional information (e.g., ECC bytes) for all the physical blocks in the logical block.
   or
b) if a physical block is being transferred, then:
   A) user data or transformed user data for all the logical blocks in the physical block;
   B) protection information or transformed protection information, if any, for all the logical blocks in the physical block; and
   C) any additional information (e.g., ECC bytes).

If a cache contains a more recent version of a logical block, the device server shall write the logical block to the medium before reading it. The values in the Read-Write Error Recovery mode page (see 4.3.15) do not apply to this command. The device server may perform retries while processing this command.

Table 118. READ LONG (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Eh)							
1	Reserved					PBLOCK	CORRCT	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved							
7	(MSB)	BYTE TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

LOGICAL BLOCK ADDRESS field

See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field.

If the additional information contain an ECC, any other additional bytes that are correctable by ECC should be included (e.g., a data synchronization mark within the area covered by ECC). It is not required for the ECC bytes to be at the end of the user data or protection information, if any; however, the ECC bytes should be in the same order as they are on the medium.

PBLOCK bit

If there is more than one logical block per physical block (i.e., the LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field in the READ CAPACITY (16) data is set to a non-zero value), then the device server shall support the physical block (PBLOCK) bit.

1 a PBLOCK bit set to one specifies that the device server shall return the entire physical block containing the specified logical block, and

0 a PBLOCK bit set to zero specifies that the device server shall return bytes representing only the specified logical block.

If there are one or more physical blocks per logical block (i.e., the LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field in the READ CAPACITY (16) data is set to zero) and the PBLOCK bit is set to one, then the device server shall return CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

CORRCT (correct) bit

0 A correct (CORRCT) bit set to zero specifies that a logical block be read without any correction made by the device server. A CORRCT bit set to zero should result in GOOD status unless data is not transferred for some reason other than that the data is non-correctable. In this case the appropriate status and sense data shall be returned.

1 A CORRCT bit set to one specifies that the data be corrected by ECC before being transferred to the data-in buffer.

BYTE TRANSFER LENGTH field

The BYTE TRANSFER LENGTH field specifies the number of bytes of data that shall be read from the specified logical block and transferred to the data-in buffer. If the BYTE TRANSFER LENGTH field is not set to zero and does not match the available data length, the device server 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. In the sense data (see SPC-4), the VALID and ILI bits shall each be set to one and the INFORMATION field shall be set to the difference (i.e., residue) of the requested byte transfer length minus the actual available data length in bytes. Negative values shall be indicated by two's complement notation.

A BYTE TRANSFER LENGTH field set to zero specifies that no bytes shall be read. This condition shall not be considered an error.

3.30 READ LONG (16) command

The READ LONG (16) command (see table 119) requests that the device server transfer data from a single logical block to the data-in buffer. The data transferred during the READ LONG (16) command is vendor-specific, but shall include the following items recorded on the medium:

a) if a logical block is being transferred, then:
   A) user data or transformed user data for the logical block;
   B) protection information or transformed protection information, if any, for the logical block; and
   C) any additional information (e.g., ECC bytes) for all the physical blocks in the logical block.
   or
b) if a physical block is being transferred, then:
   A) user data or transformed user data for all the logical blocks in the physical block;
   B) protection information or transformed protection information, if any, for all the logical blocks in the physical block; and
   C) any additional information (e.g., ECC bytes).

If a cache contains a more recent version of a logical block, the device server shall write the logical block to the medium before reading it. The values in the Read-Write Error Recovery mode page (see 4.3.15) do not apply to this command. The device server may perform retries while processing this command. This command is implemented as a service action of the SERVICE ACTION IN operation code.

Table 119. READ LONG (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Eh)							
1	Reserved			SERVICE ACTION (11h)				
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	Reserved							
11								
12	(MSB)	BYTE TRANSFER LENGTH						
13								(LSB)
14	Reserved						PBLOCK	CORRCT
15	CONTROL							

See the READ LONG (10) command (see 3.29) for the definitions of the fields in this command.

3.31 REASSIGN BLOCKS command

3.31.1 REASSIGN BLOCKS command overview

The REASSIGN BLOCKS command (see table 120) requests that the device server reassign defective logical blocks to another area on the medium set aside for this purpose. The device server should also record the location of the defective logical blocks in the GLIST, if supported. This command shall not alter the contents of the PLIST.

The parameter list provided in the data-out buffer contains a defective LBA list that contains the LBAs of the logical blocks to be reassigned. The device server shall reassign the parts of the medium used for each logical block in the defective LBA list. More than one physical block may be relocated by each LBA. If the device server is able to recover user data and protection information, if any, from the original logical block, it shall write the recovered user data and any protection information to the reassigned logical block. If the LBA is unmapped, then the device server shall cause the LBA to become mapped and write the data that was retrieved during a read operation specifying the LBA that was reassigned to the logical block (see SBC-3). If the device server is unable to recover user data and protection information, if any, it shall write vendor-specific data as the user data and shall write a default value of FFFFFFFF_FFFFFFFFh as the protection information, if enabled. The data in all other logical blocks on the medium shall be preserved.

Note. The effect of specifying a logical block to be reassigned that previously has been reassigned is to reassign the logical block again. Although not likely, over the life of the medium, a logical block may be assigned to multiple physical block addresses until no more spare locations remain on the medium.

Table 120. REASSIGN BLOCKS command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (07h)							
1	Reserved						LONGLBA	LONGLIST
2	Reserved							
4								
5	CONTROL							

LONGLBA (Long LBA) bit

0 A long LBA (LONGLBA) bit set to zero specifies that the REASSIGN BLOCKS defective LBA list contains four byte LBAs.

1 A LONGLBA bit set to one specifies that the REASSIGN BLOCKS defective LBA list contains eight byte LBAs.

3.31.2 REASSIGN BLOCKS parameter list

The REASSIGN BLOCKS parameter list (see table 121) contains a four-byte parameter list header followed by a defective LBA list containing one or more LBAs.

Table 121. REASSIGN BLOCKS parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0 – 3	Parameter list header (see table 122 or table 123)							
4 – n	DEFECTIVE LBA LIST (if any)							

LONGLIST bit

0 If LONGLIST is set to zero, the parameter list header is defined in table 122.

Table 122. REASSIGN BLOCKS short parameter list header

Bit / Byte	7	6	5	4	3	2	1	0
0 – 1	Reserved							
2 – 3	(MSB) DEFECT LIST LENGTH (LSB)							

1 If LONGLIST is set to one, the parameter list header is defined in table 123.

Table 123. REASSIGN BLOCKS long parameter list header

Bit / Byte	7	6	5	4	3	2	1	0
0 – 3	(MSB) DEFECT LIST LENGTH (LSB)							

DEFECT LIST LENGTH field

The DEFECT LIST LENGTH field indicates the total length in bytes of the DEFECTIVE LBA LIST field. The DEFECT LIST LENGTH field does not include the parameter list header length and is equal to either:

a) four times the number of LBAs, if the LONGLBA bit is set to zero; or

b) eight times the number of LBAs, if the LONGLBA bit is set to one.

DEFECTIVE LBA LIST field

The DEFECTIVE LBA LIST field contains a list of defective LBAs. Each LBA is a four-byte field if the LONGLBA bit is set to zero or an eight-byte field if the LONGLBA bit is set to one. The LBAs shall be in ascending order.

If the direct-access block device has insufficient capacity to reassign all of the specified logical blocks, the device server shall terminate the command with CHECK CONDITION status with the sense key set to HARDWARE ERROR and the additional sense code set to NO DEFECT SPARE LOCATION AVAILABLE.

If the direct-access block device is unable to successfully complete a REASSIGN BLOCKS command, the device server shall terminate the command with CHECK CONDITION status with the appropriate sense data (see SPC-4). The first LBA not reassigned shall be returned in the COMMAND-SPECIFIC INFORMATION field of the sense data. If information about the first LBA not reassigned is not available, or if all the defects have been reassigned, the COMMAND-SPECIFIC INFORMATION field shall be set to FFFFFFFFh if fixed format sense data is being used or FFFFFFFF_FFFFFFFFh if descriptor format sense data is being used.

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

If the REASSIGN BLOCKS command returns CHECK CONDITION status and the sense data COMMAND-SPECIFIC INFORMATION field contains a valid LBA, the application client should remove all LBAs from the defective LBA list prior to the one returned in the COMMAND-SPECIFIC INFORMATION field. If the sense key is MEDIUM ERROR and the INFORMATION field contains the valid LBA, the application client should insert that new defective LBA into the defective LBA list and reissue the REASSIGN BLOCKS command with the new defective LBA list. Otherwise, the application client should perform any corrective action indicated by the sense data and then reissue the REASSIGN BLOCKS command with the new defective LBA list.

3.32 RECEIVE DIAGNOSTIC RESULTS command

The RECEIVE DIAGNOSTIC RESULTS command (see table 124) requests analysis data be sent to the application client Data-In Buffer after completion of a SEND DIAGNOSTIC command (see 3.49), or, optionally, SCSI-3 Enclosure Services (SES) pages be sent to the initiator. Support for SES is indicated in the drive product manual. The drive supports the optional Page format, wherein the initiator sends additional pages after a SEND DIAGNOSTIC command. These additional pages have a page code that specifies to the drive the format of the data to be returned after it receives a RECEIVE DIAGNOSTIC RESULTS command.

If no data in the optional Page format was requested by the SEND DIAGNOSTICs command (0 in bit 4 of Table 162), the data returned to the initiator is in the format shown in Table 125.

If the SEND DIAGNOSTICs command requested either page 00h or page 40h (the only two optional pages supported by the drive), data returned is in the format shown in Table 214 or 215, respectively.

All FRU and error code definitions are unique to the product and intended for factory/field maintenance personnel.

Table 124. RECEIVE DIAGNOSTIC RESULTS command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (1Ch)							
1	Reserved							PCV
2	PAGE CODE							
3	(MSB)	ALLOCATION LENGTH						
4								(LSB)
5	CONTROL							

PCV (Page Code Valid) bit

0 A page code valid (PCV) bit set to zero specifies that the device server return parameter data based on the most recent SEND DIAGNOSTIC command (e.g., the diagnostic page with the same page code as that specified in the most recent SEND DIAGNOSTIC command). The response to a RECEIVE DIAGNOSTIC RESULTS command with the PCV bit set to zero is vendor-specific if:

a) The most recent SEND DIAGNOSTIC command was not a SEND DIAGNOSTIC command defining parameter data to return;

b) A RECEIVE DIAGNOSTIC RESULTS command with a PCV bit set to one has been processed since the last SEND DIAGNOSTIC command was processed; or

c) No SEND DIAGNOSTIC command defining parameter data to return has been processed since power on, hard reset, or logical unit reset.

1 A page code valid (PCV) bit set to one specifies that the device server return the diagnostic page specified in the PAGE CODE FIELD. Page code values are defined in Table 213.

NOTES

(a) Logical units compliant with previous versions of SPC-4 may transfer more than one diagnostic page in the parameter data if the PCV bit is set to zero and the previous SEND DIAGNOSTIC command sent more than one diagnostic page in the parameter list.

(b) To ensure that the diagnostic command information is not destroyed by a command sent from another I_T nexus, the logical unit should be reserved.

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

PAGE CODE field

The PAGE CODE field identifies which diagnostic page is requested as a result of a RECEIVE DIAGNOSTIC RESULTS command with the PCV bit equal to one, or returned as a result of a RECEIVE DIAGNOSTIC RESULTS parameter data.

ALLOCATION LENGTH field

The ALLOCATION LENGTH field (see 2.2.6) shall specify the number of bytes the initiator has allocated for returned data. An ALLOCATION LENGTH of zero indicates that no data shall be transferred. Any other value indicates the maximum number of bytes that shall be transferred. The drive terminates the DATA IN phase when ALLOCATION LENGTH bytes have been transferred or when all available data has been transferred to the initiator, whichever is less.

Table 125. Diagnostic Data Bytes

Code	Byte	Description
00h	0	ADDITIONAL LENGTH (MSB) [1]
28h	1	ADDITIONAL LENGTH (LSB) [1]
XXh	2	FRU CODE (most probable) [2]
XXh	3	FRU CODE [2]
XXh	4	FRU CODE [2]
XXh	5	FRU CODE (least probable) [2]
XXh	6	ERROR CODE (MSB) [3]
V.U.	7	ERROR CODE (LSB) [4]
V.U.	8 thru n	ADDITIONAL VENDOR UNIQUE FAULT INFORMATION (not available)

[1] ADDITIONAL LENGTH. This two byte value indicates the number of additional bytes included in the diagnostic data list. For example, if no product unique byte (byte 7) is available, this value would be 0006h. A value of 0000h means that there are no additional bytes.

[2] A FIELD REPLACEABLE UNIT (FRU) Code is a byte that identifies an assembly that may have failed. The codes will be listed in probability order, with the most probable assembly listed first and the least probable listed last. A code of 00h indicates there is no FRU information and a code of 01h indicates the entire unit should be replaced. Seagate drives return 00h in these bytes.

[3] The ERROR CODE is a two byte value that provides information designating which part of a diagnostic operation has failed. The byte 7 error code is vendor unique and defined in note [4]. Usually, Seagate drives support only some subset of the list given in note [4].

[4] Vendor Unique codes supported by Seagate devices.

Diagnostic Fault Codes

01h Formatter Diagnostic Error

02h Microprocessor RAM Diagnostic Error

04h No Drive Ready

08h No Sector or Index Detected

Diagnostic Fault Codes

09h	Fatal Hardware Error While Doing Drive Diagnostics
0Ch	No Drive Command Complete
10h	Unable to Set Drive Sector Size
14h	Unable to Clear Drive Attention
18h	Unable to Start Spindle Motor
20h	Unable to Recal Drive
30h	Unable to Send Write Current Data to Drive
34h	Unable to Issue Drive SEEK Command
40h	Unable to Read User Table From Drive
41h	Ran Out of Sectors While Doing Drive Diagnostics
42h	Unable to Read Reallocation Table
43h	Unable to Read ETF Log
45h	Firmware Read from Disc or Sent by Host has an Invalid Checksum
60h	Thermal Calibration Failure
70h	Microprocessor Internal Timer Error
80h	Buffer Controller Diagnostic Error
81h	Buffer RAM Diagnostic Error
C1h	Data Miscompare While Doing Drive Diagnostics

3.33 RELEASE(6) command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

The RELEASE(6) command (see table 126) is used to release a previously reserved logical unit. This subclause describes only those instances where the RELEASE(6) command differs from the RELEASE(10) command. Except for the instances described in this subclause, the RELEASE(6) command shall function exactly like the RELEASE(10) command (see 3.35).

Table 126. RELEASE(6) Command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (17h)							
1	Reserved			Obsolete				
2	Obsolete							
3	Reserved							
4	Reserved							
5	CONTROL							

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

Obsolete Bits 1 through 4 of Byte 1 provided a method, limited to device addresses 0 through 7, to handle third-party reservations in earlier versions of the SCSI standard. The obsolete method has been replaced by the RESERVE(10) and RELEASE(10).

Byte 1 Bit 0 and Byte 2 provide an obsolete way to release previously reserved extents within a logical unit. If Byte 1, Bit 0 is equal to one, device servers not implementing the obsolete capability shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

3.34 RELEASE (10) command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

3.34.1 RELEASE(10) command introduction

The RELEASE(10) command (see table 127) is used to release a previously reserved logical unit.

Table 127. RELEASE (10) Command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (57h)							
1	Reserved			3RDPTY	Reserved		LONGID	Obsolete
2	Obsolete							
3	THIRD-PARTY DEVICE ID							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	PARAMETER LIST LENGTH						
8								(LSB)
9	CONTROL							

The RESERVE and RELEASE commands provide a basic mechanism for contention resolution in multiple-initiator systems. See 5.5.1 for a general description of reservations and the commands that manage them. A reservation may only be released by a RELEASE command from the initiator that made it. It is not an error for an application client to attempt to release a reservation that is not currently valid, or is held by another initiator. In this case, the device server shall return GOOD status without altering any other reservation.

Byte 1 Bit 0 and Byte 2 provide an obsolete way to release previously reserved extents within a logical unit. If Byte 1, Bit 0 is equal to one, device servers not implementing the obsolete capability shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

3.34.2 Logical unit release

Logical unit reservation release is mandatory if the RELEASE(10) command is implemented. This command shall cause the device server to terminate all non-third-party logical unit reservations that are active from the initiator to the specified logical unit.

3.34.3 Third-party release

Third-party reservation release is mandatory if the RELEASE(10) command is implemented. Third-party release allows an application client to release a logical unit that was previously reserved using third-party reservation (see 7.21.3). Third-party release shall be implemented. It is intended for use in multiple-initiator systems that use the COPY and EXTENDED COPY commands.

3RDPTY (third-party) bit

0 If the third-party (3RDPTY) bit is zero, then a third-party release is not requested. If the 3RDPTY bit is zero then the LONGID bit and the PARAMETER LIST LENGTH field shall be ignored. If the 3RDPTY bit is one then the device server shall release the specified logical unit, but only if the initiator ID, 3RDPTY bit, and THIRD-PARTY DEVICE ID are identical when compared to the RESERVE command that established the reservation.

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

Note. If a target implements independent storage of mode parameters for each initiator, a third-party RESERVE command copies the current mode parameters for the initiator that sent the RESERVE command to the current mode parameters for the initiator specified as the third-party device (e.g., a copy manager SCSI device). A unit attention condition notifies the third-party of the changed mode parameters due to the reservation. A successful third-party RELEASE command does not change the third-party devices' current mode parameters back to their previous values. The third-party device may issue MODE SENSE and MODE SELECT commands to query and modify the mode parameters.

THIRD-PARTY DEVICE ID field

If the THIRD-PARTY DEVICE ID value associated with the reservation release is smaller than 255, the LONGID bit may be zero and the ID value sent in the CDB THIRD-PARTY DEVICE ID field. Device ID formats are protocol specific. If the LONGID bit is zero, the PARAMETER LIST LENGTH field shall be set to zero. If the THIRD-PARTY DEVICE ID is greater than 255, the LONGID bit shall be one.

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

If the LONGID bit is one, the parameter list length shall be eight, and the parameter list shall have the format shown in table 96. If the LONGID bit is one, the THIRD-PARTY DEVICE ID field in the CDB shall be ignored. If the LONGID bit is one and the parameter list length is not eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

Table 128. RELEASE(10) parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	THIRD-PARTY DEVICE ID							
7								

3.35 REPORT DEVICE IDENTIFIER command

See REPORT IDENTIFYING INFORMATION.

3.36 REPORT IDENTIFYING INFORMATION

The REPORT IDENTIFYING INFORMATION command requests that the device server send device identification information to the application client. The REPORT DEVICE IDENTIFYING INFORMATION command is an extension to the REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action of the MAINTENANCE IN command defined in SCC-2. Additional MAINTENANCE IN and MAINTENANCE OUT service actions are defined in SCC-2 and in this manual.

The MAINTENANCE IN service actions defined in SCC-2 shall apply only to logical units that return a device type of 0Ch or the SCCS bit set to one in their standard INQUIRY data (see 3.6.2). When a SCSI device returns a device type of 0Ch or the SCCS bit set to one in its standard INQUIRY data, the implementation requirements for the SCC-2 MAINTENANCE IN service actions shall be as specified in SCC-2. Otherwise the MAINTENANCE IN service action definitions and implementation requirements stated in this manual shall apply.

The device server shall return the same identifying information regardless of the I_T nexus being used to retrieve the identifying information.

Processing a REPORT IDENTIFYING INFORMATION command may require the enabling of a nonvolatile memory within the logical unit. If the nonvolatile memory is not ready, the command shall be terminated with CHECK CONDITION status, rather than wait for the nonvolatile memory to become ready. The sense key shall be set to NOT READY and the additional sense code shall be set as described in table (see table 30). This information should allow the application client to determine the action required to cause the device server to become ready.

Table 129. REPORT DEVICE IDENTIFIER command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (05h)				
2	Reserved							
3								
4	RESTRICTED (see SSC-3)							
5								
6	(MSB)	ALLOCATION LENGTH						
7								
8								
9								(LSB)
10	INFORMATION TYPE							Reserved
11	CONTROL							

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6.

INFORMATION TYPE field

The INFORMATION TYPE field (see table 130) specifies the type of information to be reported.

Table 130. INFORMATION TYPE

Code	Description	Length	Reference
0000000b	Peripheral device identifying information	0 to 64 bytes	
0000010b	Peripheral device text identifying information.	65 to 512bytes	
1111111b	Identifying information supported. The parameter data contains a list of supported identifying information types and the maximum length of each.	0 to 256 bytes	3.36.1
xxxxxx1b	Restricted.		SCC-2
All other	Reserved.		

The REPORT IDENTIFYING INFORMATION parameter data format used when the INFORMATION TYPE field is set to 0000000b or 0000010b is shown in table 131.

Table 131. REPORT IDENTIFYING INFORMATION parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	(MSB)	INFORMATION LENGTH (n - 3)						
3								(LSB)
4	INFORMATION							
n								

INFORMATION LENGTH

The INFORMATION LENGTH field indicates the length in bytes of the INFORMATION field. The relationship between the INFORMATION LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

INFORMATION

The INFORMATION field contains the identifying information that has the specified information type (see table 130).

3.36.1 IDENTIFYING INFORMATION SUPPORTED parameter data

The REPORT IDENTIFYING INFORMATION parameter data format used when the INFORMATION TYPE field is set to 1111111b is shown in table 132.

Table 132. REPORT IDENTIFYING INFORMATION SUPPORTED parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	(MSB)	IDENTIFYING INFORMATION LENGTH (n - 3)						
3								(LSB)
Identifying information descriptor list								
4	First Identifying information descriptor (see table 133)							
7								
	.. > >							
n - 3	Last identifying information descriptor							
n								

IDENTIFYING INFORMATION LENGTH field

The IDENTIFYING INFORMATION LENGTH field indicates the length in bytes of the identifying information descriptor list. The relationship between the IDENTIFYING INFORMATION LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

Identifying information descriptor list

The Identifying information descriptor list contains an identifying information descriptor (see table (see table 133)) for each identifying information type supported by the device server. The identifying information descriptors shall be sorted in increasing order by information type.

Table 133. Identifying Information descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	INFORMATION TYPE							Reserved
1	Reserved							
2	(MSB)	MAXIMUM INFORMATION LENGTH						
3								(LSB)

INFORMATION TYPE

The INFORMATION TYPE field indicates the information type (see table 130).

MAXIMUM LENGTH

The MAXIMUM LENGTH field indicates the maximum number of bytes supported for identifying information that has the indicated information type.

3.37 REPORT LUNS command

The REPORT LUNS command (see table 134) requests that the peripheral device logical unit inventory accessible to the I_T nexus be sent to the application client. The logical unit inventory is a list that shall include the logical unit numbers of all logical units having a PERIPHERAL QUALIFIER value of 000b (see table 49). Logical unit numbers for logical units with PERIPHERAL QUALIFIER values other than 000b and 011b may be included in the logical unit inventory. Logical unit numbers for logical units with a PERIPHERAL QUALIFIER value of 011b shall not be included in the logical unit inventory.

Table 134. REPORT LUNS command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A0h)							
1	Reserved							
2	SELECT REPORT							
3	Reserved							
4								
5								
6	(MSB)	ALLOCATION LENGTH						
7								
8								
9								(LSB)
10	Reserved							
11	CONTROL							

SELECT REPORT field

The SELECT REPORT field (see table 135) specifies the types of logical unit addresses that shall be reported.

Table 135. SELECT REPORT field

Code	Description
00h	The list shall contain the logical units accessible to the I_T nexus with the following addressing methods (see SAM-4): a Logical unit addressing method, b Peripheral device addressing method; and c Flat space addressing method. If there are no logical units, the LUN LIST LENGTH field shall be zero.
01h	The list shall contain only well known logical units, if any. If there are no well known logical units, the LUN LIST LENGTH field shall be zero.
02h	The list shall contain all logical units accessible to the I_T nexus.
03h - FFh	Reserved

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6. The allocation length should be at least 16.

Note. Device servers compliant with SPC return CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB when the allocation length is less than 16 bytes.

The REPORT LUNS command shall return CHECK CONDITION status only when the device server is unable to return the requested report of the logical unit inventory.

If a REPORT LUNS command is received from an I_T nexus with a pending unit attention condition (i.e., before the device server reports CHECK CONDITION status), the device server shall perform the REPORT LUNS command (see SAM-4).

The REPORT LUNS parameter data should be returned even though the device server is not ready for other commands. The report of the logical unit inventory should be available without incurring any media access delays. If the device server is not ready with the logical unit inventory or if the inventory list is null for the requesting I_T nexus and the SELECT REPORT field set to 02h, then the device server shall provide a default logical unit inventory that contains at least LUN 0 or the REPORT LUNS well known logical unit. A non-empty peripheral device logical unit inventory that does not contain either LUN 0 or the REPORT LUNS well known logical unit is valid.

If a REPORT LUNS command is received for a logical unit that the SCSI target device does not support and the device server is not capable of returning the logical unit inventory, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to LOGICAL UNIT NOT SUPPORTED.

The device server shall report those devices in the logical unit inventory using the format shown in table 136.

Table 136. REPORT LUNS parameter data format

Byte \ Bit	7	6	5	4	3	2	1	0
0	(MSB)							
	LUN LIST LENGTH (N-7)							
3								(LSB)
4								
	Reserved							
7								
	LUN LIST							
8								
	FIRST LUN							
15								
	...							
n-7								
	LAST LUN							
n								

LUN LIST LENGTH field

The LUN LIST LENGTH field shall contain the length in bytes of the LUN list that is available to be transferred. The LUN list length is the number of logical unit numbers in the logical unit inventory multiplied by eight. The relationship between the LUN LIST LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

3.38 REPORT SUPPORTED OPERATION CODES command

3.38.1 Introduction

The REPORT SUPPORTED OPERATION CODES command requests information on commands the addressed logical unit supports. An application client may request a list of all operation codes and service actions supported by the logical unit or the command support data for a specific command.

The REPORT SUPPORTED OPERATION CODES command is a service action of the MAINTENANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and in this manual. The MAINTENANCE IN service actions defined in SCC-2 apply only to logical units that return a device type of 0Ch or the SCCS bit set to one in their standard INQUIRY data.

Table 137. REPORT SUPPORTED OPERATION CODES command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (0Ch)				
2	RCTD	Reserved				REPORTING OPTIONS		
3	REQUESTED OPERATION CODE							
4	(MSB)	REQUESTED SERVICE ACTION						
5								(LSB)
6	(MSB)	ALLOCATION LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

RCTD (Return Command Timeouts Descriptor)

0 A RCTD bit set to zero specifies that the command timeouts descriptor shall not be included in any parameter data returned.

1 A RCTD bit set to one specifies that the command timeouts descriptor shall be included in each command descriptor that is returned or in the one-command parameter that is returned.

REPORTING OPTIONS field

The REPORTING OPTIONS field specifies the information to be returned in the parameter data.

Code	Descriptions	Parameter Data Reference
000b	A list of all operation codes and service actions supported by the logical unit shall be returned in the all_commands parameter data format. The REQUESTED OPERATION CODE CDB field and REQUESTED SERVICE ACTION CDB field shall be ignored.	3.38.2
001b	The command support data for the operation code specified in the REQUESTED OPERATION CODE field shall be returned in the one_command parameter data format. The REQUESTED SERVICE ACTION CDB field shall be ignored. If the REQUESTED OPERATION CODE field specifies an operation code that has service actions, then the 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.	3.38.3
010b	The command support data for the operation code and service action specified in the REQUESTED OPERATION CODE CDB field and REQUESTED SERVICE ACTION CDB field shall be returned in the one_command parameter data format. If the REQUESTED OPERATION CODE CDB field specifies an operation code that does not have service actions, then the 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.	3.38.4
011b - 111b	Reserved.	

REQUESTED OPERATION CODE field

The REQUESTED OPERATION CODE field specifies the operation code of the command to be returned in the one_command parameter data format (see 3.38.3).

REQUESTED SERVICE ACTION field

The REQUESTED SERVICE ACTION field specifies the service action of the command to be returned in the one_command parameter data format.

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6.

3.38.2 All-commands parameter data format

The REPORT SUPPORTED OPERATION CODES all_commands parameter data format begins with a four-byte header that contains the length in bytes of the parameter data followed by a list of supported commands. Each command descriptor contains information about a single supported command CDB (i.e., one operation code and service action combination, or one non-service-action operation code). The list of command descriptors shall contain all commands supported by the logical unit.

Table 138. All_commands parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	COMMAND DATA LENGTH (n - 3)						
3								(LSB)
	Command descriptors							
4	Command descriptor 0							
	. . .							
	Command descriptor x							
n								

The COMMAND DATA LENGTH field indicates the length in bytes of the command descriptor list.

Each command descriptor (see table 139) contains information about a single supported command CDB.

Table 139. Command descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE							
1	Reserved							
2	(MSB)	SERVICE ACTION						
3								(LSB)
4	Reserved							
5	Reserved						CTDP	SERVACTV
6	(MSB)	CDB LENGTH						
7								(LSB)
8	Command timeouts descriptor, if any							
19								

OPERATION CODE field

The OPERATION CODE field contains the operation code of a command supported by the logical unit.

SERVICE ACTION field

The SERVICE ACTION field contains a supported service action of the supported operation code indicated by the OPERATION CODE field. If the operation code indicated in the OPERATION CODE field does not have a service action, the SERVICE ACTION field shall be set to 00h.

CTDP (Command Timeouts Descriptor Present) bit

0 A CTDP bit set to zero indicates that the command timeouts descriptor is not included in this command descriptor.

1 A CTDP bit set to one indicates that the command timeouts descriptor is included in this command descriptor.

SERVACTV (Service Action Valid) bit

0 A service action valid (SERVACTV) bit set to zero indicates the operation code indicated by the OPERATION CODE field does not have service actions and the SERVICE ACTION field contents are reserved.

1 A SERVACTV bit set to one indicates the operation code indicated by the OPERATION CODE field has service actions and the contents of the SERVICE ACTION field are valid.

CDB LENGTH field

The CDB LENGTH field contains the length of the command CDB in bytes for the operation code indicated in the OPERATION CODE field, and if the SERVACTV bit is set to the service action indicated by the SERVICE ACTIONS field.

RCTD bit

0 If the RCTD bit is set to zero, the command timeouts descriptor shall not be included.

1 If the RCTD bit is set to one in the REPORT SUPPORTED OPERATION CODES CDB, the command timeouts descriptor shall be included.

3.38.3 One_command parameter data format

The REPORT SUPPORTED OPERATION CODES one_command parameter data format contains information about the CDB and a usage map for bits in the CDB for the command specified by the REPORTING OPTIONS, REQUESTED OPERATION CODE, and REQUESTED SERVICE ACTION fields in the REPORT SUPPORTED OPERATION CODES CDB.

Table 140. One_command parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	CTDP	Reserved				SUPPORT		
2	(MSB)	CDB SIZE (n - 3)						
3								(LSB)
4	CDB USAGE DATA							
n								
n+1	Command timeouts descriptor, if any							
n+12								

CTDP bit

0 A CTDP bit set to zero indicates that the command timeouts descriptor is not included in the parameter data.

1 A CTDP bit set to one indicates that the command timeouts descriptor is included in the parameter data.

SUPPORT field

The SUPPORT field is defined in table 141.

Table 141. SUPPORT values

Support	Description
000b	Data about the requested SCSI command is not currently available. All data after byte 1 is not valid. A subsequent request for command support data may be successful.
001b	The device server does not support the requested command. All data after byte 1 is undefined.
010b	Reserved.
011b	The device server supports the requested command in conformance with a SCSI standard. The parameter data format conforms to the definition in table 140.
100b	Reserved.
101b	The device server supports the requested command in a vendor specific manner. The parameter data format conforms to the definition in table 140.
110b - 111b	Reserved.

CDB SIZE field

The CDB SIZE field contains the size of the CDB USAGE DATA field in the parameter data, and the number of bytes in the CDB for command being queried (i.e., the command specified by the REPORTING OPTIONS, REQUESTED OPERATION CODE, and REQUESTED SERVICE ACTION fields in the REPORT SUPPORTED OPERATION CODES CDB).

CDB USAGE DATA field

The CDB USAGE DATA field contains information about the CDB for the command being queried. The first byte of the CDB USAGE DATA field shall contain the operation code for the command being queried. If the command being queried contains a service action, then that service action code shall be placed in the CDB USAGE DATA field in the same location as the SERVICE ACTION field of the command CDB. All other bytes of the CDB USAGE DATA field shall contain a usage map for bits in the CDB for the command being queried.

The bits in the usage map shall have a one-for-one correspondence to the CDB for the command being queried. If the device server evaluated a bit in the CDB for the command being queried, the usage map shall contain a one in the corresponding bit position. If any bit representing part of a field is returned as one, all bits for the field shall be returned as one. If the device server ignores or treats as reserved a bit in the CDB for the command being queried, the usage map shall contain a zero in the corresponding bit position. The usage map bits for a given CDB field all shall have the same value.

FOr example, the CDB usage bit map for the REPORT SUPPORTED OPERATION CODES command is: A3h, 0Ch, 87h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, 00h, 07h. This example assumes that the logical unit only supports the low-order three bits of the CDB CONTROL byte. The first byte contains the operation code, and the second byte contains three reserved bits and the service action. The remaining bytes contain the usage map.

RCTD bit

0 If the RCTD bit is set to zero, the command timeouts descriptor shall not be included.

1 If the RCTD bit is set to one in the REPORT SUPPORTED OPERATION CODES CDB, the command timeouts descriptor shall be included.

3.38.4 Command timeouts descriptor

3.38.4.1 Overview

The command timeouts descriptor (see table 142) returns timeout information for commands supported by the logical unit based on the time from the start of processing for the command to its reported completion.

Values returned in the command timeouts descriptor do not include times that are outside the control of the device server (e.g., prior commands with the IMMED bit set to one in the CDB, concurrent commands from the same or different I_T nexuses, manual unloads, power-on self tests, prior aborted commands, commands that force cache synchronization, delays in the service delivery subsystem).

For commands that cause a change in power condition, values returned in the command timeouts descriptor do not include the power condition transition time (e.g., the time to spinup rotating media).

Values returned in the command timeouts descriptor should not be used to compare products.

Table 142. Command timeouts descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	DESCRIPTOR LENGTH (OAh)						
1								(LSB)
2	Reserved							
3	COMMAND SPECIFIC							
4	(MSB)	NOMINAL COMMAND PROCESSING TIMEOUT						
7								(LSB)
8	(MSB)	RECOMMEND COMMAND TIMEOUT						
11								(LSB)

DESCRIPTOR LENGTH field

The DESCRIPTOR LENGTH field indicates the number of bytes that follow in the command timeouts descriptor.

COMMAND SPECIFIC field

The COMMAND SPECIFIC field contains timeout information (see table 143) that is specific to one or more commands. If no command specific timeout information is defined by this or the applicable command standard the COMMAND SPECIFIC field is reserved.

Table 143. Command timeout descriptor COMMAND SPECIFIC field usage in this manual

Command	Reference
WRITE BUFFER	3.38.4.2

NOMINAL COMMAND PROCESSING TIMEOUT field

A non-zero value in the NOMINAL COMMAND PROCESSING TIMEOUT field indicates the minimum amount of time in seconds the application client should wait prior to querying for the progress of the command identified by the parameter data that contains this command timeouts descriptor. A value of zero in the NOMINAL COMMAND PROCESSING TIME-OUT field indicates that no timeout is indicated.

Note. The value contained in the NOMINAL COMMAND PROCESSING TIMEOUT field may include time required for typical device error recovery procedures expected to occur on a regular basis.

RECOMMENDED COMMAND TIMEOUT field

A non-zero value in the RECOMMENDED COMMAND TIMEOUT field specifies the recommended time in seconds the application client should wait prior to timing out the command identified by the parameter data that contains this command timeouts descriptor. A value of zero in the RECOMMENDED COMMAND TIMEOUT field indicates that no time is indicated.

The device server should set the recommended command timeout to a value greater than or equal to the nominal command processing timeout.

3.38.4.2 WRITE BUFFER command timeouts descriptor COMMAND SPECIFIC field usage

For the WRITE BUFFER command, the COMMAND SPECIFIC field usage is reserved for all modes except the following:

a. Download microcode mode (04h);
b. Download microcode and save mode (05h);
c. Download microcode with offsets mode (06h);
d. Download microcode with offsets and save mode (07h);
e. Download microcode with offsets and defer activation mode (0Eh) only if the microcode is activated by an event other than an activate deferred microcode mode; and
f. Activate deferred microcode mode (0Fh).

If the command timeouts descriptor describes one of the WRITE BUFFER modes listed in this subclause, then the COMMAND SPECIFIC field indicates the maximum time, in one second increments, that access to the SCSI device is limited or not possible through any SCSI ports associated with a logical unit that processes a WRITE BUFFER command that specifies one of the named modes. A value of zero in the COMMAND SPECIFIC field indicates that the no maximum time is indicated.

3.39 REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS commands

The REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS command requests information on task management functions (see SAM-4) the addressed logical unit supports.

The REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS command is a service action of the MAINTENTANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and this manual. The MAINTENANCE IN service actions defined in SCC-2 apply only to logical units that return a device type of OCh or the SCCS bit set to one in their standard INQUIRY data (see 3.6.2).

Table 144. REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (0Dh)				
2 5	Reserved							
6 9	(MSB)	ALLOCATION LENGTH (4h or larger)						(LSB)
10	Reserved							
11	CONTROL							

ALLOCATION LENGTH field

The ALLOCATION LENGTH field (see 2.2.6) specifies the number of bytes that have been allocated for the returned parameter data. The allocation length should be at least four. If the allocation length is less than four, the 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.

The format of the parameter data returned by the REPORT TASK MANAGEMENT FUNCTIONS command is shown in table 145.

Table 145. REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	ATS	ATSS	CACAS	CTSS	LURS	QTS	TRS	WAKES
1	Reserved					QAES	QTSS	ITNRS
2 3	Reserved							

ATS (ABORT TASK SUPPORTED) bit

1 An ATS bit set to one indicates the ABORT TASK task management functions (see SAM-4) is supported by the logical unit

0 An ATS bit set to zero indicates the ABORT TASK task management function is not supported.

ATSS (ABORT TASK SET) bit

1 ATS bit set to one indicates the ABORT TASK SET task management function (see SAM-4) is supported by the logical unit.

0 ATS bit set to zero indicates the ABORT TASK SET task management function is not supported.

CACAS (CLEAR ACA bit

1 A CACAS bit set to one indicates the CLEAR ACA task management function (see SAM-4) is supported by the logical unit.

0 A CACAS bit set to zero indicates the CLEAR ACA task management function is not supported.

CTSS (CLEAR TASK SET) bit

1 A CTSS bit set to one indicates the CLEAR TASK SET task management function (see SAM-4) is supported by the logical unit.

0 A CTSS bit set to zero indicates the CLEAR TASK SET task management function is not supported.

LURS (LOGICAL UNIT RESET) bit

1 A LURS bit set to one indicates the LOGICAL UNIT RESET task management function (see SAM-4) is supported by the logical unit.

0 A LURS bit set to zero indicates the LOGICAL UNIT RESET task management function is not supported.

QTS (QUERY TASK) bit

1 A QTS bit set to one indicates the QUERY TASK task management function (see SAM-4) is supported by the logical unit.

0 A QTS bit set to zero indicates the QUERY TASK task management function is not supported.

TRS (TARGET RESET) bit

1 A TRS bit set to one indicates the TARGET RESET task management function (See SAM-2) is supported by the logical unit.

0 A TRS bit set to zero indicates the TARGET RESET task management function is not supported.

WAKES (WAKEUP) bit

1 A WAKES bit set to one indicates the WAKEUP task management function (see SAM-2) is supported by the logical unit.

0 A WAKES bit set to zero indicates the WAKEUP task management function is not supported.

QAES (QUERY ASYNCHRONOUS EVENT supported) bit

1 A QAES bit set to one indicates the QUERY ASYNCHRONOUS EVENT task management function (see SAM-4) is supported by the logical unit.

0 A QAES bit set to zero indicates the QUERY ASYNCHRONOUS EVENT task management function is not supported.

QTSS (QUERY TASK SET supported) bit

1 A QUERY TASK SET supported (QTSS) bit set to one indicates the QUERY TASK SET task management function (see SAM-4) is supported by the logical unit.

0 A QTSS bit set to zero indicates the QUERY TASK SET task management function is not supported.

ITNRS (I-T NEXUS RESET) bit

1 An ITNRS bit set to one indicates the I_T NEXUS RESET task management function (see SAM-4) is supported by the logical unit.

0 An ITNRS bit set to zero indicates the I_T NEXUS RESET task management function is not supported.

3.40 REPORT TIMESTAMP command

The REPORT TIMESTAMP command requests that the device server return the value of the logical unit's timestamp.

The REPORT TIMESTAMP command is a service action of the MAINTENANCE IN command. Additional MAINTENANCE IN service actions are defined in SCC-2 and in this standard. The MAINTENANCE IN service actions defined in SCC-2 apply only to logical units that return a device type of 0Ch or the SCCS bit set to one in their standard INQUIRY data (see 3.6.2).

Table 146. REPORT TIMESTAMP command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A3h)							
1	Reserved			SERVICE ACTION (0Fh)				
2 – 5	Reserved							
6 – 9	(MSB) ALLOCATION LENGTH (LSB)							
10	Reserved							
11	CONTROL							

ALLOCATION LENGTH field

The ALLOCATION LENGTH field is defined in 2.2.6.

The format for the parameter data returned by the REPORT TIMESTAMP command is shown in table 147.

Table 147. REPORT TIMESTAMP parameter data format

Bit / Byte	7	6	5	4	3	2	1	9
0 – 1	(MSB) TIMESTAMP PARAMETER DATA LENGTH (0Ah) (LSB)							
2	Reserved					TIMESTAMP ORIGIN		
3	Reserved							
4 – 9	TIMESTAMP							
10	Reserved							
11	Rerserved							

TIMESTAMP DATA LENGTH field

The TIMESTAMP DATA LENGTH field indicates the number of bytes of parameter data that follow. The relationship between the TIMESTAMP PARAMETER DATA LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

TIMESTAMP ORIGIN field

The TIMESTAMP ORIGIN field indicates the origin of the timestamp (see SPC-4).

TIMESTAMP field

The TIMESTAMP field contains the current value of the timestamp (see SPC-4).

3.41 REQUEST SENSE command

The REQUEST SENSE command (see table 148) requests that the device server transfer sense data to the application client.

Table 148. REQUEST SENSE command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (03h)							
1	Reserved							DESC
2	Reserved							
3								
4	ALLOCATION LENGTH							
5	CONTROL							

DESC (Descriptor Format) bit

The descriptor format (DESC) bit specifies which sense data format shall be returned.

0 If DESC is set to zero, fixed format sense data shall be returned (see 2.4.1.2).

1 If DESC is set to one and descriptor format sense data is supported, descriptor format sense data shall be returned (see 2.4.1.1).

Note. If DESC is set to one and the descriptor format sense data is not supported, Seagate drives reject the bit.

ALLOCATION LENGHT field

The ALLOCATION LENGTH field is defined in 2.2.6. Application clients should request 252 bytes of sense data to ensure they retrieve all the sense data. If fewer than 252 bytes are requested, sense data may be lost since the REQUEST SENSE command with any allocation length clears the sense data.

CONTROL field

The CONTROL field is described in clause 2.2.7.

Sense data shall be available and cleared under the conditions defined in SAM-4. If the device server has no other sense data available to return, it shall:

1) return the sense key set to NO SENSE, additional sense code set to NO ADDITIONAL SENSE INFORMATION and;
2) complete the REQUEST SENSE command with GOOD status.

If the logical unit is in an idle power condition (see SPC-4), the device server shall process a REQUEST SENSE command by:

1) returning parameter data containing sense data with the sense key set to NO SENSE and the additional sense code set to one of the following:
   A) LOW POWER CONDITION ON if the reason for entry into the idle power condition is unknown;
   B) IDLE CONDITION ACTIVATED BY TIMER if the logical unit entered the idle_a power condition due to the idle_a condition timer (see SPC-4);
   C) IDLE CONDITION ACTIVATED BY COMMAND if the logical unit entered the idle_a power condition due to receipt of a command requiring the idle_a power condition while it was in a power condition with a higher processing priority order (see SPC-4);
   D) IDLE_B CONDITION ACTIVATED BY TIMER if the logical unit entered the idle_b power condition due to the idle_b condition timer (see SPC-4);
   E) IDLE_B CONDITION ACTIVATED BY COMMAND if the logical unit entered the idle_b power condition due to receipt of a command requiring the idle_b power condition while it was in a power condition with a higher processing priority order (see SPC-4);
   F) IDLE_C CONDITION ACTIVATED BY TIMER if the logical unit entered the idle_c power condition due to the idle_c condition timer (see SPC-4); or
   G) IDLE_C CONDITION ACTIVATED BY COMMAND if the logical unit entered the idle_c power condition due to receipt of a command requiring the idle_c power condition while it was in a power condition with a higher processing priority order (see SPC-4);

   and
2) complete the REQUEST SENSE command with GOOD status.

If the logical unit is in a standby power condition (see SPC-4), the device server shall process a REQUEST SENSE command by:

1) return parameter data containing sense data with the sense key set to NO SENSE and the additional sense code set to one of the following:
   A) LOW POWER CONDITION ON if the reason for entry into the standby power condition is unknown;
   B) STANDBY_Y CONDITION ACTIVATED BY TIMER if the logical unit entered the standby_y power condition due to the standby_y condition timer (see SPC-4);
   C) STANDBY_Y CONDITION ACTIVATED BY COMMAND if the logical unit entered the standby_y power condition due to receipt of a command requiring the standby_y power condition while it was in the standby_z power condition; or
   D) STANDBY CONDITION ACTIVATED BY TIMER if the logical unit entered the standby_z power condition due to the standby_z condition timer (see SPC-4);

   and
2) complete the REQUEST SENSE command with GOOD status.

On completion of the command the logical unit shall return to the same power condition that was active before the REQUEST SENSE command was received. A REQUEST SENSE command shall not reset any power condition timers.

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

Examples of conditions that cause a REQUEST SENSE command to return a CHECK CONDITION status are:

a) An invalid field value is detected in the CDB;
b) The device server does not support the REQUEST SENSE command (see 3.41);
c) An unrecovered error is detected by the service delivery subsystem; or
d) A malfunction prevents return of the sense data.

If a REQUEST SENSE command is received on an I_T nexus with a pending unit attention condition (i.e., before the device server reports CHECK CONDITION status) and there is an exception condition specific to the REQUEST SENSE command itself, then the device server shall not clear the pending unit attention condition (see SAM-4).

If a recovered error occurs during the processing of the REQUEST SENSE command, the device server shall:

1) return parameter data containing sense data with the sense key set to RECOVERED ERROR; and
2) complete the REQUEST SENSE command with GOOD status.

In response to a REQUEST SENSE command issued to a logical unit that reports a peripheral qualifier of 011b in its standard INQUIRY data (see 3.6.2) the device server shall:

1) return parameter data containing sense data with the sense key set to ILLEGAL REQUEST and the additional sense code shall be set to LOGICAL UNIT NOT SUPPORTED; and
2) complete the REQUEST SENSE command with GOOD status.

In response to a REQUEST SENSE command issued to a logical unit that reports a peripheral qualifier of 001b in its standard INQUIRY data, the device server shall:

1) return parameter data containing sense data with the sense key set to ILLEGAL REQUEST and the additional sense code shall be set to LOGICAL UNIT NOT SUPPORTED; and
2) complete the REQUEST SENSE command with GOOD status.

In response to a REQUEST SENSE command issued to a logical unit that reports a peripheral qualifier of 000b in its standard INQUIRY data because it has a peripheral device connected but is not ready for access, the device server shall:

1) return parameter data containing sense data appropriate to the condition that is making the logical unit not operational; and
2) complete the REQUEST SENSE command with GOOD status.

In response to a REQUEST SENSE command issued to a logical unit that reports a peripheral qualifier of 000b in its standard INQUIRY data because the device server is unable to determine whether or not a peripheral device is connected, the device server shall:

1) return parameter data containing sense data with the sense key set to NO SENSE and the additional sense code set to NO ADDITIONAL SENSE INFORMATION; and
2) complete the REQUEST SENSE command with GOOD status.

Device servers shall return at least 18 bytes of parameter data in response to a REQUEST SENSE command if the allocation length is 18 or greater and the DESC bit is set to zero. Application clients may determine how much sense data has been returned by examining the ALLOCATION LENGTH field in the CDB and the ADDITIONAL SENSE LENGTH field in the sense data. Device servers shall not adjust the additional sense length to reflect truncation if the allocation length is less than the sense data available.

3.42 RESERVE(6) command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

The RESERVE(6) command (see table 149) is used to reserve a logical unit. This subclause describes only those instances where the RESERVE(6) command differs from the RESERVE(10) command. Except for the instances described in this subclause, the RESERVE(6) command shall function exactly like the RESERVE(10) command (see 3.40).

Table 149. RESERVE(6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (16h)							
1	Reserved			Obsolete				
2	Obsolete							
3	Obsolete							
4								
5	CONTROL							

Obsolete Bits 1 through 4 of Byte 1 provided a method, limited to device addresses 0 through 7, to handle third-party reservations in earlier versions of the SCSI standard. The obsolete method has been replaced by the RESERVE(10) and RELEASE(10).

Byte 1 Bit 0 and Bytes 2 through 4 provide an obsolete way to reserve extents within a logical unit. The Seagate device server does not support the feature--if it is set, it will be rejected with an ILLEGAL REQUEST/ INVALID BIT IN THE CDB. The Seagate device server ignores the Obsolete field for the Reservation ID.

3.43 RESERVE (10) command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

3.43.1 RESERVE (10) command introduction

The RESERVE(10) command (see table 150) is used to reserve a logical unit.

Table 150. RESERVE (10) Command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (56h)							
1	Reserved			3RDPTY	Reserved		LONGID	Obsolete
2	Obsolete							
3	THIRD-PARTY DEVICE ID							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	PARAMETER LIST LENGTH						
8								(LSB)
9	CONTROL							

The RESERVE and RELEASE commands provide a basic mechanism for contention resolution in multiple-initiator systems. The third-party reservation allows logical units to be reserved for another specified SCSI device. See 5.5.1 for a general description of reservations and the commands that manage them.

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

Byte 1 Bit 0 and Byte 2 provide an obsolete way to reserve extents within a logical unit. If Byte 1, Bit 0 is equal to one, device servers not implementing the obsolete capability shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST.

3.43.2 Logical unit reservation

Logical unit reservations are mandatory if the RESERVE(10) command is implemented. This command shall request that the entire logical unit be reserved for the exclusive use of the initiator until the reservation is superseded by another valid RESERVE command from the same initiator or until released by a RELEASE command from the same initiator that made the reservation, by a TARGET RESET task management function performed by any initiator, by a hard reset condition, or by a power on cycle. A logical unit reservation shall not be granted if the logical unit is reserved by another initiator. It shall be permissible for an initiator to reserve a logical unit that is currently reserved by that initiator. If the LONGID bit or the 3RDPTY bit is zero then the PARAMETER LIST LENGTH field shall be ignored.

If the logical unit is reserved for another initiator, the device server shall return RESERVATION CONFLICT status. After honoring a logical unit reservation, the device server shall check each newly received command for reservation conflicts.

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

3.43.3 Third-party reservation

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

0 If the third-party (3RDPTY) bit is zero, then a third-party reservation is not requested. If the 3RDPTY bit is zero then the LONGID bit shall be ignored.

1 If the 3RDPTY bit is one then the device server shall reserve the specified logical unit for the SCSI device specified in the THIRD-PARTY DEVICE ID field. Device ID formats are protocol specific. The device server shall preserve the reservation until it is superseded by another valid RESERVE command from the initiator that made the reservation or until it is released by the same initiator, by a TARGET RESET task management function performed by any initiator, a hard reset condition, or by a power on cycle. The device server shall ignore any attempt to release the reservation made by any other initiator.

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

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

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

THIRD-PARTY DEVICE ID field

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

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

If the LONGID bit is one, the parameter list length shall be eight, and the parameter list shall have the format shown in table 151. If the LONGID bit is one and the parameter list length is not eight, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST.

Table 151. RESERVE(10) ID only parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	THIRD-PARTY DEVICE ID							
7								

3.43.4 Superseding reservations

Superseding reservations is mandatory if the RELEASE(10) command is implemented. An application client that holds a current logical unit reservation may modify that reservation by issuing another RESERVE command to the same logical unit. The superseding RESERVE command shall release the previous reservation state when the new reservation request is granted. The current reservation shall not be modified if the superseding reservation request is not granted. If the superseding reservation cannot be granted because of conflicts with a previous reservation, other than the reservation being superseded, then the device server shall return RESERVATION CONFLICT status.

Note. Superseding reservations allow the SCSI device ID in a third-party reservation to be changed. This capability is necessary for certain situations when using the EXTENDED COPY command.

3.44 REZERO UNIT command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

Table 152. REZERO UNIT command (01h)

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (01h)							
1	LUN [1]			LOGICAL BLOCK ADDRESS MUST BE 0'S				
2	LOGICAL BLOCK ADDRESS MUST BE 0'S							
3								
4	Reserved							
5	CONTROL [2]							

The Rezero Unit command (see table 152) requests that the disc drive set its logical block address to zero and return the disc drive read/write heads to the track (or cylinder) containing Logical Block Zero. This command is intended for systems which disable retries and the initiator performs error recovery. It is longer than a seek to Logical Block Address zero and should be utilized if seek errors are encountered.

For systems that support disconnection, the disc drive disconnects when this command is received.

A Rezero Command also causes a thermal compensation to occur and resets the thermal compensation cycle timer back to its start, thus allowing the host to know when to expect the next thermal compensation to occur. The host can thus prevent critical data transfer operations from being interrupted at an undesirable time.

For drives that support saved log parameters, the Rezero Unit command will also save log counters to the media and reset the log save timer back to its start.

Note.

[1] The LUN must be zero.

[2] See “Control Byte” 2.2.7.

3.45 SECURITY PROTOCOL IN command

3.45.1 Command description

The SECURITY PROTOCOL IN command (see table 153)is used to retrieve security protocol information or the results of one or more SECURITY PROTOCOL OUT commands.

Table 153. SECURITY PROTOCOL IN command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A2h)							
1	SECURITY PROTOCOL							
2	SECURITY PROTOCOL SPECIFIC							
3								
4	INC_512	Reserved						
5	Reserved							
6	(MSB)	ALLOCATION LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

SECURITY PROTOCOL field

The SECURITY PROTOCOL field (see table 154) specifies which security protocol is being used.

Table 154. SECURITY PROTOCOL field in SECURITY PROTOCOL IN command

Code	Description	Reference
00h	Security protocol information	
01h - 06h	Defined by the TCG	https://www.trustedcomputinggroup.org
07h	CbCS	SPC-4
08h - 1Fh	Reserved	
20h	Tape Data Encryption	SSC-3
21h	Data Encryption Configuration	ADC-3
22h - 3Fh	Reserved	
40h	SA Creation Capabilities	SPC-4
41h	IKEv2-SCSI	SPC-4

Table 154. SECURITY PROTOCOL field in SECURITY PROTOCOL IN command

Code	Description	Reference
42h to ECh	Reserved	
EDh	SDcard TrustedFlash Security Systems Specification 1.1.3	
EEh	Authentication in Host Attachments of Transient Storage Deices	IEEE 1667
EFh	ATA Device Server Password Security	
F0h - FFh	Vendor specific	

Note. The SECURITY PROTOCOL SPECIFIC field code values 22h - 2Fh are tentatively reserved for SSC-x uses.

The contents of the SECURITY PROTOCOL SPECIFIC field depend on the protocol specified by the SECURITY PROTOCOL field (see table 154).

INC_512 (512 Increment)

0 An INC_512 bit set to zero specifies that the ALLOCATION LENGTH field (see 2.2.6) expresses the number of bytes to be transferred.

1 A 512 increment (INC_512) bit set to one specifies that the ALLOCATION LENGTH field (see 2.2.6) expresses the maximum number of bytes available to receive data in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1,024 bytes, etc.). Pad bytes may or may not be appended to meet this length. Pad bytes shall have a value of 00h.

Indications of data overrun or underrun and the mechanism, if any, for processing retries depend on the protocol specified by the SECURITY PROTOCOL field (see table 154).

Any association between a previous SECURITY PROTOCOL OUT command and the data transferred by a SECURITY PROTOCOL IN command depends on the protocol specified by the SECURITY PROTOCOL field (see table 155). If the device server has no data to transfer (e.g., the results for any previous SECURITY PROTOCOL OUT commands are not yet available), the device server may transfer data indicating it has no other data to transfer.

The format of the data transferred depends on the protocol specified by the SECURITY PROTOCOL field (see table 154).

The device server shall retain data resulting from a SECURITY PROTOCOL OUT command, if any, until one of the following events is processed:

a. Transfer of the data via a SECURITY PROTOCOL IN command from the same I_T_L nexus as defined by the protocol specified by the SECURITY PROTOCOL field;
b. Logical unit reset (See SAM-4); or
c. I_T nexus loss (See SAM-4) associated with the I_T nexus that sent the SECURITY PROTOCOL OUT command.

If the data is lost due to one of these events the application client may send a new SECURITY PROTOCOL OUT command to retry the operation.

3.45.2 Security protocol information description

3.45.2.1 Overview

The purpose of security protocol information security protocol (i.e., the SECURITY PROTOCOL field set to 00h in a SECURITY PROTOCOL IN command) is to transfer security protocol related information from the logical unit. A SECURITY PROTOCOL IN command in which the SECURITY PROTOCOL field is set to 00h is not associated with a previous SECURITY PROTOCOL OUT command and shall be processed without regard for whether a SECURITY PROTOCOL OUT command has been processed.

If the SECURITY PROTOCOL IN command is supported, the SECURITY PROTOCOL value of 00h shall be supported as defined in this standard.

3.45.2.2 CDB description

When the SECURITY PROTOCOL field is set to 00h in a SECURITY PROTOCOL IN command, the SECURITY PROTOCOL SPECIFIC field contains a single numeric value as defined in 155.

Table 155. SECURITY PROTOCOL SPECIFIC field for SECURITY PROTOCOL IN protocol 00h

Code	Description	Support	Reference
0000h	Supported security protocol list	Mandatory	
0001h	Certificate data	Mandatory	
0002h - FFFFh	Reserved		

All other CDB fields for SECURITY PROTOCOL IN command shall meet the requirements stated in 3.45.1.

Each time a SECURITY PROTOCOL IN command with the SECURITY PROTOCOL field set to 00h is received, the device server shall transfer the data defined in 3.45.2 starting with byte 0.

3.45.2.3 Supported security protocols list description

If the SECURITY PROTOCOL field is set to 00h and the SECURITY PROTOCOL SPECIFIC field is set to 0000h in a SECURITY PROTOCOL IN command, the parameter data shall have the format shown in table 156.

Table 156. Supported security protocols SECURITY PROTOCOL IN parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
5								
6	(MSB) SUPPORTED SECURITY PROTOCOL LIST LENGTH (m - 7)							
7								(LSB)
8	SUPPORTED SECURITY PROTOCOL (first) (00h)							
	. . .							
m	SUPPORTED SECURITY PROTOCOL (last)							
m + 1	Pad bytes (optional)							
n								

SUPPORTED SECURITY PROTOCOL LIST LENGTH field

The SUPPORTED SECURITY PROTOCOL LIST LENGTH field indicates the total length, in bytes, of the supported security protocol list that follows.

SUPPORTED SECURITY PROTOCOL field

Each SUPPORTED SECURITY PROTOCOL field in the supported security protocols list shall contain one of the security protocol values supported by the logical unit. The values shall be listed in ascending order starting with 00h.

ALLOCATION LENGTH field

The total data length shall conform to the ALLOCATION LENGTH field requirements (see 2.2.6). Pad bytes may be appended to meet this length. Pad bytes shall have a value of 00h.

3.45.2.4 Certificate data description

3.45.2.4.1 Certificate overview

A certificate is either an X.509 Public Key Certificate or an X.509 Attribute Certificate depending on the capabilities of the logical unit.

If the SECURITY PROTOCOL field is set to 00h and the SECURITY PROTOCOL SPECIFIC field is set to 0001h in a SECURITY PROTOCOL IN command, the parameter data shall have the format shown in table 157.

Table 157. Certificate data SECURITY PROTOCOL IN parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	(MSB)	CERTIFICATE LENGTH (m - 3)						
3								(LSB)
4	CERTIFICATE							
m								
m + 1	Pad bytes (optional)							
n								

CERTIFICATE LENGTH field

The CERTIFICATE LENGTH field indicates the total length, in bytes, of the certificate or certificates that follow. The length may include more than one certificates. If the device server doesn't have a certificate to transfer, the CERTIFICATE LENGTH field shall be set to 0000h.

CERTIFICATE field

The contents of the CERTIFICATE field are defined in 3.45.2.4.2 and 3.45.2.4.3.

ALLOCATION LENGTH field

The total data length shall conform to the ALLOCATION LENGTH field requirements (see 2.2.6). Pad bytes may be appended to meet this length. Pad bytes shall have a value of 00h.

3.45.2.4.2 Public Key certificate description

RFC 3280 defines the certificate syntax for certificates consistent with X.509v3 Public Key Certificate Specification. Any further restrictions beyond the requirements of RFC 3280 are yet to be defined by T10.

3.45.2.4.3 Attribute certificate description

FRC 3281 defines the certificate syntax for certificates consistent with X.509v2 Attribute certificate Specification. Any further restrictions beyond the requirements of RFC 3281 are yet to be defined by T10.

3.46 SECURITY PROTOCOL OUT command

The SECURITY PROTOCOL OUT command (see table 158) is used to send data to the logical unit. The data sent specifies one or more operations to be performed by the logical unit. The format and function of the operations depends on the contents of the SECURITY PROTOCOL field(see table 159). Depending on the protocol specified by the SECURITY PROTOCOL field, the application client may use the SECURITY PROTOCOL IN command (see 3.45) to retrieve data derived from these operations.

Table 158. SECURITY PROTOCOL OUT command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (B5h)							
1	SECURITY PROTOCOL							
2	SECURITY PROTOCOL SPECIFIC							
3								
4	INC_512	Reserved						
5	Reserved							
6	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

SECURITY PROTOCOL field

The SECURITY PROTOCOL field (see table 159) specifies which security protocol is being used.

Table 159. SECURITY PROTOCOL field in SECURITY PROTOCOL OUT command

Code	Description	Reference
00h	Reserved	
01h - 06h	Defined by the TCG	https://www.trustedcomputinggroup.org
07h	CbCS	SPC-4
08h - 1Fh	Reserved	
20h	Tape Data Encryption	SSC-3
21h	Data Encryption Configuration	ADC-3
22h - 40h	Reserved	
41h	IKEv2-SCSI	SPC-4
22h - 40h	Reserved	

Table 159. SECURITY PROTOCOL field in SECURITY PROTOCOL OUT command

Code	Description	Reference
EDh	SDcard TrustedFlash Security Systems Specification	
EEh	Authentication in Host Attachments of Transient Storage Devices	IEEE 1667
EFh	ATA Device Server Password Security	
F0h - FFh	Vendor specific	

Note. The SECURITY PROTOCOL SPECIFIC field code values 21h - 2Fh are tentatively reserved for SSC-x uses.

SECURITY PROTOCOL SPECIFIC field

The contents of the SECURITY PROTOCOL SPECIFIC field depend on the protocol specified by the SECURITY PROTOCOL field (see table 159).

INC_512 (512 Increment)

0 A INC_512 bit set to zero specifies that the TRANSFER LENGTH field indicates the number of bytes to be transferred.

1 INC_512 bit set to one specifies that the TRANSFER LENGTH field (see 2.2.4) expresses the number of bytes to be transferred in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1,024 bytes, etc.). Pad bytes shall be appended as needed to meet this requirement. Pad bytes shall have a value of 00h.

Any association between a SECURITY PROTOCOL OUT command and a subsequent SECURITY PROTOCOL IN command depends on the protocol specified by the SECURITY PROTOCOL field (see table 159). Each protocol shall define whether:

a. The device server shall complete the command with GOOD status as soon as it determines the data has been correctly received. An indication that the dat has been processed is obtained by sending a SECURITY PROTOCOL IN command and receiving the results in the associated data transfer; or

b. The device server shall complete the command with GOOD status only after the data has been successfully processed and an associated SECURITY PROTOCOL IN command is not required.

The format of the data transferred depends on the protocol specified by the SECURITY PROTOCOL field (see table 159).

3.47 SEEK command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

Table 160. SEEK command (0Bh)

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0bh)							
1	LUN [1]			MSB				
2	LOGICAL BLOCK ADDRESS [2]							
3								LSB
4	Reserved							
5	CONTROL [3]							

The Seek command (see table 160) requests that the disc drive seek to the specified logical block address.

For systems which support disconnection, the disc drive disconnects when a valid Seek command is received. The use of this command is infrequent since all commands involving data transfer to/from the disc drive media contain implied seek addresses.

Note.

[1] The Logical Unit number should be zero.

[2] The maximum Logical Block Address that may be specified for a Seek command is defined in Read

Capacity Data.

[3] See “Control Byte” (see 2.2.7).

3.48 SEEK EXTENDED command

This command has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

Table 161. SEEK command (2Bh)

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (2bh)							
1	LUN [1]			Reserved				
2	LOGICAL BLOCK ADDRESS (MSB)							
3	LOGICAL BLOCK ADDRESS							LSB
4	LOGICAL BLOCK ADDRESS							
5	LOGICAL BLOCK ADDRESS (LSB) [2]							
6	Reserved							
7	Reserved							
8	Reserved							
9	CONTROL [3]							

The Seek Extended command (see table 161) requests that the disc drive seek to the specified logical block address.

This command is implemented with the disc drive specific parameters listed in Table 161.

LUN (Logical Unit Number) field

Seagate drives only support Logical Unit Number 0.

Note. Byte 1 should be 00h.

LOGICAL BLOCK ADDRESS field

Four byte Logical Address may be specified. The drive will seek to this address when the command is received.

Note.

[1] In the CDB, the LUN must be zero.

[2] The maximum Logical Block Address that may be specified for a Seek command is defined in Read Capacity Data.

[3] See “Control Byte” (see 2.2.7).

3.49 SEND DIAGNOSTIC command

The SEND DIAGNOSTIC command (see table 162) requests the device server to perform diagnostic operations on the SCSI target device, on the logical unit, or on both. Logical units that support this command shall implement, at a minimum, the default self-test feature (i.e., the SELFTEST bit equal to one and a parameter list length of zero).

Table 162. SEND DIAGNOSTIC command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (1Dh)							
1	SELF-TEST CODE			PF	Reserved	SELFTEST	DEVOFFL	UNITOFFL
2	Reserved							
3	(MSB) PARAMETER LIST LENGTH							
4								(LSB)
5	CONTROL							

SELFTEST bit

1 If the SELFTEST bit is set to one, the SELF-TEST CODE field shall contain 000b.

0 If the SELFTEST bit is set to zero, the contents of SELF-TEST CODE field are specified in table 163.

SELF-TEST CODE field

Table 163. SELF-TEST CODE field

Code	Name	Description
000b		This value shall be used when the SELFTEST bit is set to one, or when the SELFTEST bit is set to zero and the PF bit is set to one.
001b	Background short self-test	The device server shall start its short self-test (see SPC-4) in the background mode (see SPC-4). The PARAMETER LIST LENGTH field shall contain zero.
010b	Background extended self-test	The device server shall start its extended self-test (see SPC-4) in the background mode (see SPC-4). The PARAMETER LIST LENGTH field shall contain zero.
011b	Reserved	
100b	Abort background self-test	The device server shall abort the current self-test running in background mode. The PARAMETER LIST LENGTH field shall contain zero. This value is only valid if a previous SEND DIAGNOSTIC command specified a background self-test function and that self-test has not completed. If either of these conditions is not met, the 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.
101b	Foreground short self-test	The device server shall start its short self-test (see SPC-4) in the foreground mode (see SPC-4). The PARAMETER LIST LENGTH field shall contain zero.
110b	Foreground extended self-test	The device server shall start its extended self-test (see SPC-4) in the foreground mode (see SPC-4). The PARAMETER LIST LENGTH field shall contain zero.
111b	Reserved	

PF (page format) bit

1 A page format (PF) bit set to one specifies that the SEND DIAGNOSTIC parameters and any parameters returned by a following RECEIVE DIAGNOSTIC RESULTS command with the PCV bit set to zero shall contain a single diagnostic page as defined in 4.1.1.

Note. Logical units compliant with previous versions of SPC-4 may transfer more than one diagnostic page in the SEND DIAGNOSTIC command's parameter list and by doing so may request that more than one diagnostic page be transmitted in the RECEIVE DIAGNOSTIC RESULTS command's parameter data.

0 A PF bit set to zero specifies that all SEND DIAGNOSTIC parameters are vendor specific. If the PARAMETER LIST LENGTH field is set to zero and the SEND DIAGNOSTIC command is not going to be followed by a corresponding RECEIVE DIAGNOSTIC RESULTS command with the PCV bit set to zero, then the application client shall set the PF bit to zero. The implementation of the PF bit is optional.

SELFTEST (Self-test) bit

1 A self-test (SELFTEST) bit set to one specifies that the device server shall perform the logical unit default self-test. If the self-test successfully passes, the command shall be terminated with GOOD status. If the self-test fails, the command shall be terminated with CHECK CONDITION status, with the sense key set to HARDWARE ERROR.

0 A SELFTEST bit set to zero specifies that the device server shall perform the diagnostic operation specified by the SELF-TEST CODE field or in the parameter list. The diagnostic operation may require the device server to return parameter data that contains diagnostic results. If the return of parameter data is not required, the return of GOOD status indicates successful completion of the diagnostic operation. If the return of parameter data is required, the device server shall either:

[a] Perform the requested diagnostic operation, prepare the parameter data to be returned and indicate completion by returning GOOD status. The application client issues a RECEIVE DIAGNOSTIC RESULTS command to recover the parameter data; or

[b] Accept the parameter list, and if no errors are detected in the parameter list, return GOOD status. The requested diagnostic operation and the preparation of the parameter data to be returned are performed upon receipt of a RECEIVE DIAGNOSTIC RESULTS command.

UNITOFFL (unit offline) bit

1 A unit offline (UNITOFFL) bit set to one specifies that the device server may perform diagnostic operations that may affect the user accessible medium on the logical unit (e.g., write operations to the user accessible medium, or repositioning of the medium on sequential access devices). The device server may ignore the UNITOFFL bit.

0 A UNITOFFL bit set to zero specifies that, after the device server has completed any diagnostic operations specified by the SEND DIAGNOSTIC command, the user accessible medium shall exhibit no effects resulting from the device server's processing the SEND DIAGNOSTIC command that are detectable by any application client. When the SELFTEST bit is set to zero, the UNITOFFL bit shall be ignored.

DEVOFFL (SCSI target device offline) bit

1 A SCSI target device offline (DEVOFFL) bit set to one grants permission to the device server to perform diagnostic operations that may affect all the logical units in the SCSI target device (e.g., alteration of reservations, log parameters, or sense data). The device server may ignore the DEVOFFL bit.

0 A DEVOFFL bit set to zero specifies that, after the device server has completed any diagnostic operations specified by the SEND DIAGNOSTIC command, no logical unit shall exhibit any effects resulting from the device server's processing the SEND DIAGNOSTIC command that are detectable by any application client. If the SELFTEST bit is set to zero, the DEVOFFL bit shall be ignored.

PARAMETER LIST LENGTH field

The PARAMETER LIST LENGTH field specifies the length in bytes of the parameter list that shall be transferred from the application client Data-Out Buffer to the device server. A parameter list length of zero specifies that no data shall be transferred. This condition shall not be considered an error. If PF bit is set to one and the specified parameter list length results in the truncation of the diagnostic page (e.g., the parameter list length does not match the page length specified in the diagnostic page), then the 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.

To ensure that the diagnostic command information is not destroyed by a command sent from another I_T nexus, the logical unit should be reserved.

3.50 SET IDENTIFYING INFORMATION command

The SET IDENTIFYING INFORMATION command (see table 202) requests that the device server set identifying information in the logical unit to the value received in the SET IDENTIFYING INFORMATION parameter list. The SET IDENTIFYING INFORMATION command is an extension to the SET PERIPHERAL DEVICE/ COMPONENT DEVICE IDENTIFIER service action of the MAINTENANCE OUT command defined in SCC-2. Additional MAINTENANCE IN and MAINTENANCE OUT service actions are defined in SCC-2 and in this standard.

The MAINTENANCE OUT service actions defined only in SCC-2 shall apply only to logical units that return a device type of 0Ch (i.e., storage array controller device) or the SCCS bit set to one in their standard INQUIRY data. When a logical unit returns a device type of 0Ch or the SCCS bit set to one in its standard INQUIRY data, the implementation requirements for the SCC-2 MAINTENANCE OUT service actions shall be as specified in SCC-2. Otherwise the MAINTENANCE OUT service action definitions and implementation requirements stated in this manual shall apply.

On successful completion of a SET IDENTIFYING INFORMATION command that changes identifying information saved by the logical unit, the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus on which the SET IDENTIFIER command was received, with the additional sense code set to DEVICE IDENTIFIER CHANGED.

Table 164. SET IDENTIFYING INFORMATION command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (A4h)							
1	Reserved			SERVICE ACTION (06h)				
2	Reserved							
3								
4	RESTRICTED (see SSC-2)							
5								
6	(MSB)	PARAMETER LIST LENGTH						
9								(LSB)
10	INFORMATION TYPE							Reserved
11	CONTROL							

The PARAMETER LIST LENGTH field specifies the length in bytes of the identifying information that shall be transferred from the application client to the device server. A parameter list length of zero specifies that no data shall be transferred, and that subsequent REPORT IDENTIFYING INFORMATION commands shall return the INFORMATION LENGTH field set to zero for the specified information type.

The INFORMATION TYPE specifies the identifying information type to be set.

Code	Description
0000000b	Peripheral device identifying information. If the PARAMETER LIST LENGTH field is set to greater than the maximum length of the peripheral device identifying information, the device server 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.
0000010b	Peripheral device text identifying information (see 3.36). If the PARAMETER LIST LENGTH field is set to a value greater than the maximum length of the peripheral device text identifying information (see 3.36), the device server 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 format of the INFORMATION field is incorrect, the device server 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.
xxxxxx1b	Restricted (see SCC-2)
All other	Reserved.

The SET IDENTIFYING INFORMATION parameter list (see table 165) contains the identifying information to be set by the device server.

Table 165. SET IDENTIFYING INFORMATION parameter list

Bit / Byte	7	6	5	4	3	2	1	9
0		INFORMATION						
n								

INFORMATION

The INFORMATION field specifies the identifying information to be set for the specified information type (see 3.36).

Upon successful completion of a SET IDENTIFYING INFORMATION command, the identifying information that is saved by the logical unit shall persist through logical unit resets, hard resets, power loss, I_T nexus losses, media format operations, and media replacement.

3.51 SET TIMESTAMP command

The SET TIMESTAMP command (see table 166) requests the device server to initialize the timestamp. If the SCSIP bit is set to one or the TCMOS bit is set to one in the Control Extension mode page. If the SCSIP bit is set to zero, the SET TIMESTAMP 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.

The SET TIMESTAMP command is a service action of the MAINTENANCE OUT command. Additional MAINTENANCE OUT service actions are defined in SCC-2 and in this standard. The MAINTENANCE OUT service actions defined only in SCC-2 apply only to logical units that return a device type of 0Ch or the SCCS bit set to one in their standard INQUIRY data (see 3.6.2).

Table 166. SET TIMESTAMP command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0Fh)							
1	Reserved			SERVICE ACTION (0Fh)				
2	Reserved							
5								
6	(MSB)	PARAMETER LIST LENGTH						
9								(LSB)
10	Reserved							
11	CONTROL							

PARAMETER LIST LENGTH field

The PARAMETER LIST LENGTH field specifies the length in bytes of the SET TIMESTAMP parameters that shall be transferred from the application client to the device server. A parameter list length of zero indicates that no data shall be transferred, and that no change shall be made to the timestamp.

The format for the parameter data returned by the SET TIMESTAMP command is shown in table 167.

Table 167. SET TIMESTAMP parameter data format

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
3								
4	TIMESTAMP							
9								
10	Reserved							
11	Reserved							

TIMESTAMP field

The TIMESTAMP field shall contain the initial value of the timestamp. The timestamp should be the number of milliseconds that have elapsed since midnight, 1 January 1970 UT. If the high order byte in the TIMESTAMP field is greater than F0h, the 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 PARAMETER LIST.

On successful completion of a SET TIMESTAMP command the device server shall generate a unit attention condition for the initiator port associated with every I_T nexus except the I_T nexus on which the SET TIMESTAMP command was received (see SAM-4), with the additional sense code set to TIMESTAMP CHANGED.

3.52 START STOP UNIT command

The START STOP UNIT command (see table 168) requests that the device server change the power condition of the logical unit or load or eject the medium. This includes specifying that the device server enable or disable the direct-access block device for medium access operations by controlling power conditions and timers.

Logical units that contain cache shall write all cached logical blocks to the medium (e.g., as they would do in response to a SYNCHRONIZE CACHE command (see 3.53 and 3.54) with the SYNC_NV bit set to zero, the LOGICAL BLOCK ADDRESS field set to zero, and the NUMBER OF BLOCKS field set to zero) prior to entering into any power condition that prevents accessing the medium (e.g., before the rotating media spindle motor is stopped during transition to the stopped power condition).

Table 168. START STOP UNIT command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (1Bh)							
1	Reserved							IMMED
2	Reserved							
3	Reserved				POWER CONDITION MODIFER			
4	POWER CONDITION				Reserved	NO_ FLUSH	LOEJ	START
5	CONTROL							

IMMED (Immediate) bit

0 If the immediate (IMMED) bit is set to zero, then the device server shall return status after the operation is completed.

1 If the IMMED bit set to one, then the device server shall return status as soon as the CDB has been validated.

POWER CONDITION field and POWER CONDITION MODIFIER FIELD

The combinations of values in the POWER CONDITION field and POWER CONDITION MODIFIER FIELD used to specify that the logical unit be placed into a power condition or to adjust a timer as defined in table 169. If this field is supported and is set to a value other than 0h, then the START and LOEJ bits shall be ignored.

Table 169. POWER CONDITION field and POWER CONDITION MODIFIER field

POWER CONDITION	POWER CONDITION Name	POWER CONDITION MODIFIER value	Description
0h	START_VALID	0h	Process the START and LOEJ bits.
1h	ACTIVE	0h	Cause the logical unit to transition to the active power condition (see SPC-4).
2h	IDLE	0h	Cause the logical unit to transition to the idle_a power condition (see SPC-4).
		1h	Cause the logical unit to transition to the idle_b power condition (see SPC-4).
			Cause the logical unit to transition to the idle_c power condition (see SPC-4).
3h	STANDBY	0h	Cause the logical unit to transition to the standby_z power condition (see SPC-4).
		1h	Cause the logical unit to transition to the standby_y power condition (see SPC-4).
5h	Obsolete	0h to Fh	Obsolete
7h	LU_CONTROL	0h	Initialize and start all of the idle condition timers that are enabled (see SPC-4), and initialize and start all of the standby condition timers that are enabled (see SPC-4).
Ah	FORCE_IDLE_0	0h	Force the idle_a condition timer to be set to zero (see SPC-4).
		1h	Force the idle_b condition timer to be set to zero (see SPC-4).
		2h	Force the idle_c condition timer to be set to zero (see SPC-4).
Bh	FORCE_STANDBY_0	0h	Force the standby_z condition timer to be set to zero (see SPC-4).
		1h	Force the standby_y condition timer to be set to zero (see SPC-4).
All other combinations			Reserved

If the START STOP UNIT command specifies a power condition that conflicts with an operation in progress (e.g., a background self test), then, after the START STOP UNIT command completes with GOOD status, the logical unit may not be in the power condition that was requested by the command.

It is not an error to specify that the logical unit transition to its current power condition.

NO_FLUSH bit

0 If the NO_FLUSH bit is set to zero, then logical units that contain cache shall write all cached logical blocks to the medium (e.g., as they would do in response to a SYNCHRONIZE CACHE command (see 3.53 and 3.54) with the SYNC_NV bit set to zero, the LOGICAL BLOCK ADDRESS field set to zero, and the NUMBER OF LOGICAL BLOCKS field set to zero) prior to entering into any power condition that prevents accessing the medium (e.g., before the rotating media spindle motor is stopped during transition to the stopped power condition).

1 If the NO_FLUSH bit is set to one, then cached logical blocks should not be written to the medium by the logical unit prior to entering into any power condition that prevents accessing the medium.

LOEJ (load eject) bit

0 If the load eject (LOEJ) bit is set to zero, then the logical unit shall take no action regarding loading or ejecting the medium.

1 If the LOEJ bit is set to one, then the logical unit shall unload the medium if the START bit is set to zero. If the LOEJ bit is set to one, then the logical unit shall load the medium if the START bit is set to one. If the POWER CONDITION field is supported and is set to a value other than 0h, then the device server shall ignore the LOEJ bit.

START bit

0 If the START bit is set to zero, then the device server shall:

a) cause the logical unit to transition to the stopped power condition;
b) stop any idle condition timer that is enabled (see SPC-4); and
c) stop any standby condition timer that is enabled (see SPC-4).

1 If the START bit set to one, then the device server shall:

1) comply with requirements defined in SCSI transport protocol standards (e.g., the NOTIFY (ENABLE SPINUP) requirement (see SAS-2));
2) cause the logical unit to transition to the active power condition;
3) initialize and start any idle condition timer that is enabled; and
4) initialize and start any standby condition timer that is enabled.

POWER CONDITION field

If the POWER CONDITION field is supported and is set to a value other than 0h, then the device server shall ignore the START bit

3.53 .SYNCHRONIZE CACHE (10) command

The SYNCHRONIZE CACHE (10) command (see table 170) requests that the device server ensure that the specified logical blocks have their most recent data values recorded in non-volatile cache and/or on the medium, based on the SYNC_NV bit. Logical blocks include user data and, if the medium is formatted with protection information enabled, protection information. Logical blocks may or may not be removed from volatile cache and non-volatile cache as a result of the synchronize cache operation.

Table 170. SYNCHRONIZE CACHE (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (35h)							
1	Reserved					SYNC_NV	IMMED	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved			GROUP NUMBER				
7	(MSB)	NUMBER OF BLOCKS						
8								(LSB)
9	CONTROL							

LOGICAL BLOCK ADDRESS field

See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field.

GROUP NUMBER field

See 2.2.8 for the definition of the GROUP NUMBER field.

SYNC_NV bit

The SYNC_NV bit (see table 171) specifies whether the device server is required to synchronize volatile and non-volatile caches.

Table 171. SYNC_NV bit

Code	Device server requirement to synchronize logical blocks currently in the	
	Volatile cache	Non-volatile cache
0	Device server shall synchronize to the medium.	Device server shall synchronize to the medium.
1	If a non-volatile cache is present, device server shall synchronize to non-volatile cache or the medium. If a non-volatile cache is not present, device server shall synchronize to the medium.	No requirement.

IMMED (Immediate) bit

0 An immediate (IMMED) bit set to zero specifies that the device server shall not return status until the operation has been completed.

1 An IMMED bit set to one specifies that the device server shall return status as soon as the CDB has been validated. If the IMMED bit is set to one and the device server does not support the IMMED bit, the device server 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.

NUMBER OF BLOCKS field

The NUMBER OF BLOCKS field specifies the number of logical blocks that shall be synchronized, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A NUMBER OF BLOCKS field set to zero specifies that all logical blocks starting with the one specified in the LOGICAL BLOCK ADDRESS field to the last logical block on the medium shall be synchronized. If the logical block address plus the number of blocks exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

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

3.54 SYNCHRONIZE CACHE (16) command

The SYNCHRONIZE CACHE (16) command (see table 172) requests that the device server ensure that the specified logical blocks have their most recent data values recorded in non-volatile cache and/or on the medium, based on the SYNC_NV bit. Logical blocks include user data and, if the medium is formatted with protection information enabled, protection information. Logical blocks may or may not be removed from volatile cache and non-volatile cache as a result of the synchronize cache operation.

Table 172. SYNCHRONIZE CACHE (16) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (91h)							
1	Reserved					SYNC_NV	IMMED	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	NUMBER OF BLOCKS						
13								(LSB)
14	Reserved			GROUP NUMBER				
15	CONTROL							

See the SYNCHRONIZE CACHE (10) command (see 3.53) for the definitions of the fields in this command.

3.55 TEST UNIT READY command

The TEST UNIT READY command (see table 173) provides a means to check if the logical unit is ready. This is not a request for a self-test. If the logical unit is able to accept an appropriate medium-access command without returning CHECK CONDITION status, this command shall return a GOOD status. If the logical unit is unable to become operational or is in a state such that an application client action (e.g., START UNIT command) is required to make the logical unit ready, the command shall be terminated with CHECK CONDITION status, with the sense key set to NOT READY.

Table 173. TEST UNIT READY command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (00h)							
1 – 4	Reserved							
5	CONTROL							

Table 174 defines the suggested GOOD and CHECK CONDITION status responses to the TEST UNIT READY command. Other conditions, including deferred errors, may result in other responses (e.g., BUSY or RESERVATION CONFLICT status).

Table 174. Preferred TEST UNIT READY responses

Status	Sense Key	Additional Sense Code
GOOD	not applicable	not applicable
CHECK CONDITION	ILLEGAL REQUEST	LOGICAL UNIT NOT SUPPORTED
CHECK CONDITION	NOT READY	LOGICAL UNIT DOES NOT RESPOND TO SELECTION
CHECK CONDITION	NOT READY	MEDIUM NOT PRESENT
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, CAUSE NOT REPORTABLE
CHECK CONDITION	NOT READY	LOGICAL UNIT IS IN PROCESS OF BECOMING READY
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, MANUAL INTERVENTION REQUIRED
CHECK CONDITION	NOT READY	LOGICAL UNIT NOT READY, FORMAT IN PROGRESS

3.56 UNMAP command

3.56.1 UNMAP command overview

The UNMAP command (see table 175) requests that the device server cause one or more LBAs to be unmapped. The UNMAP command is one of the possible commands that shall be implemented by device servers supporting thin provisioning (see SBC-3).

Table 175. UNMAP command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (42h)							
1	Reserved							
.....								
5								
6	Reserved			GROUP NUMBER				
7	(MSB) PARAMETER LIST LENGTH							
8								(LSB)
9	CONTROL							

OPERATION CODE field

The OPERATION CODE field is defined in 2.2.1 and shall be set to the value defined in table 175.

GROUP NUMBER field

See the PRE-FETCH (10) command (see 3.17) for the definition of the GROUP NUMBER field.

PARAMETER LIST LENGTH field

The PARAMETER LIST LENGTH field specifies the length in bytes of the UNMAP parameter data that shall be sent from the application client to the device server. A PARAMETER LIST LENGTH set to zero specifies that no data shall be sent.

CONTROL byte

The contents of the CONTROL byte are defined in SAM-4.

3.56.2 UNMAP parameter list

The UNMAP parameter list (see table 176) contains the data sent by an application client along with an UNMAP command. Included in the data are an UNMAP parameter list header and block descriptors for LBA extents to be processed by the device server for the UNMAP command. The LBAs specified in the block descriptors may contain overlapping extents, and may be in any order.

For each specified LBA:

a) the mapped LBA should be unmapped (see SBC-3), or may remain mapped; and
b) the unmapped LBA shall remain unmapped.

Table 176. UNMAP parameter list

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
.....				UNMAP parameter list header (see table 177)				
7								(LSB)
8	(MSB)							
.....				UNMAP block descriptor data (see table 178)				
n								(LSB)

The UNMAP parameter list header is defined in table 177.

Table 177. UNMAP parameter list header

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			UNMAP DATA LENGTH (n-1)				
1								(LSB)
2	(MSB)			UNMAP BLOCK DESCRIPTOR DATA LENGTH (n-7)				
3								(LSB)
4								
.....				Reserved				
7								

UNMAP DATA LENGTH field

The UNMAP DATA LENGTH field specifies the length in bytes of the following data that is available to be transferred from the data-out buffer. The unmap data length does not include the number of bytes in the UNMAP DATA LENGTH field.

UNMAP BLOCK DESCRIPTOR DATA LENGTH field

The UNMAP BLOCK DESCRIPTO DATA LENGTH field specifies the length in bytes of the UNMAP block descriptor data (see table 178) that is available to be transferred from the data-out buffer. If the UNMAP BLOCK DESCRIPTOR DATA

LENGTH is set to zero, then no unmap block descriptors are included in the UNMAP parameter data. This condition shall not be considered an error.

Table 178. UNMAP block descriptor data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
.....	UNMAP block descriptor (first) (see table 179)							
15								(LSB)
.....	.....							
n-15	(MSB)							
.....	UNMAP block descriptor (last) (see table 179)							
n								(LSB)

Table 179 defines an UNMAP block descriptor.

Table 179. UNMAP block descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
.....	UNMAP LOGICAL BLOCK ADDRESS							
7								(LSB)
8	(MSB)							
....	NUMBER OF LOGICAL BLOCKS							
11								(LSB)
12	Reserved							
.....								
15								

UNMAP LOGICAL BLOCK ADDRESS

The UNMAP LOGICAL BLOCK ADDRESS field contains the first LBA of the UNMAP block descriptor to be unmapped.

NUMBER OF LOGICAL BLOCKS field

The NUMBER OF LOGICAL BLOCKS field contains the number of LBAs to be unmapped beginning with the LBA specified by the UNMAP LOGICAL BLOCK ADDRESS field.

If the NUMBER OF LOGICAL BLOCKS is set to zero, then no LBAs shall be unmapped for this UNMAP block descriptor. This condition shall not be considered an error.

If the LBA specified by the UNMAP LOGICAL BLOCK ADDRESS field plus the number of logical blocks exceeds the capacity of the medium, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

If the total number of logical blocks specified in the UNMAP block descriptor data exceeds the value indicated in the MAXIMUM UNMAP LBA COUNT field in the Block Limits VPD page (see 4.4.3), or if the number of UNMAP block descriptors exceeds the value of the MAXIMUM UNMAP PARAMETER COUNT field in the Block Limits VPD page, then the device server 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.

3.57 VERIFY (10) command

The VERIFY (10) command (see table 180) requests that the device server verify the specified logical block(s) on the medium. Each logical block includes user data and may include protection information, based on the VRPROTECT field and the medium format.

Table 180. VERIFY (10) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Fh)							
1	VRPROTECT			DPO	Reserved		BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	RESTRICTED FOR MMC-5	Reserved		GROUP NUMBER				
7	(MSB)	VERIFICATION LENGTH						
8								(LSB)
9	CONTROL							

Logical units that contain cache shall write referenced cached logical blocks to the medium for the logical unit (e.g., as they would do in response to a SYNCHRONIZE CACHE command (see 3.53 and 3.54) with the SYNC_NV bit set to zero, the LOGICAL BLOCK ADDRESS field set to the value of the VERIFY command's LOGICAL BLOCK ADDRESS field, and the NUMBER OF BLOCKS field set to the value of the VERIFY command's VERIFICATION LENGTH field).

See the READ (10) command (see 3.20) for the definition of the DPO bit. See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field. See 2.2.8 for the definition of the GROUP NUMBER field.

If the Verify Error Recovery mode page (see 4.3.20) is implemented, then the current settings in that page specify the verification criteria. If the Verify Error Recovery mode page is not implemented, then the verification criteria is vendor-specific.

BYTCHK bit

0 If the byte check (BYTCHK) bit is set to zero, the device server shall:

a) perform a medium verification with no data comparison and not transfer any data from the data-out buffer; and

b) check protection information read from the medium based on the VRPROTECT field as described in table 181.

1 If the BYTCHK bit is set to one, the device server shall:

a) perform a byte-by-byte comparison of user data read from the medium and user data transferred from the data-out buffer;

b) check protection information read from the medium based on the VRPROTECT field as described in table 182;

c) check protection transferred from the data-out buffer based on the VRPROTECT field as described in table 183; and

d) perform a byte-by-byte comparison of protection information read from the medium and transferred from the data-out buffer based on the VRPROTECT field as described in table 184.

The order of the user data and protection information checks and comparisons is vendor-specific.

If a byte-by-byte comparison is unsuccessful for any reason, the device server shall terminate the command with CHECK CONDITION status with the sense key set to MISCOMPARE and the additional sense code set to the appropriate value for the condition.

VERIFICATION LENGTH field

The VERIFICATION LENGTH field specifies the number of contiguous logical blocks that shall be verified, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. If the BYTCHK bit is set to one, the VERIFICATION LENGTH field also specifies the number of logical blocks that the device server shall transfer from the data-out buffer. A VERIFICATION LENGTH field set to zero specifies that no logical blocks shall be verified. This condition shall not be considered as an error. Any other value specifies the number of logical blocks that shall be verified. If the logical block address plus the verification length exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The VERIFICATION LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page.

If the BYTCHK bit is set to zero, the device server shall check the protection information read from the medium based on the VRPROTECT field as described in table 181.

Table 181. VRPROTECT field with BYTCHK set to zero - medium (Sheet 1 of 3)

Code	Logical unit formatted with protection information	Field in protection information [g]	Extended INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
000b	Yes	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLICATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REFERENCE TAG	REF_CHK = 1 [h]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	No protection information on the medium to check. Only user data is checked.		

Table 181. VRPROTECT field with BYTCHK set to zero - medium (Sheet 2 of 3)

Code	Logical unit formatted with protection information	Field in protection information [g]	Extended INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
001b 101b [b]	Yes	LOGICAL BLOCK GUARD	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		LOGICAL BLOCK APPLI-CATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REF-ERENCE TAG	REF_CHK = 1 [h]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	Error condition [a]		
010b [b]	Yes	LOGICAL BLOCK GUARD	No check performed	
		LOGICAL BLOCK APPLI-CATION TAG	APP_CHK = 1 [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
			APP_CHK = 0	No check performed
		LOGICAL BLOCK REF-ERENCE TAG	REF_CHK = 1 [h]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
			REF_CHK = 0	No check performed
	No	Error condition [a]		
011b [b]	Yes	LOGICAL BLOCK GUARD	No check performed	
		LOGICAL BLOCK APPLI-CATION TAG	No check performed	
		LOGICAL BLOCK REF-ERENCE TAG	No check performed	
	No	Error condition [a]		

Table 181. VRPROTECT field with BYTCHK set to zero - medium (Sheet 3 of 3)

Code	Logical unit formatted with protection information	Field in protection information [g]	Extended INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code
100b [b]	Yes	logical block guard	GRD_CHK = 1	LOGICAL BLOCK GUARD CHECK FAILED
			GRD_CHK = 0	No check performed
		logical block application tag	No check performed	
		logical block reference tag	No check performed	
	No	Error condition [a]		
101b - 111b	Reserved			

a A verify operation to a logical unit that supports protection information and has not been formatted with protection information 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.

b If the logical unit does not support protection information the requested command should 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.

c The device server shall check the logical block application tag if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the VERIFY (32) command (see 3.60) is used and the ATO bit is set to one in the Control mode page (see SPC-4), this knowledge is acquired from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge may be obtained by a method not defined by this manual.

d If an error is reported, the sense key shall be set to ABORTED COMMAND.

e If multiple errors occur, the selection of which error to report is not defined by this manual.

f See the Extended INQUIRY Data VPD page (see SPC-4) for the definitions of the GRD_CHK bit, the APP_CHK bit, and the REF_CHK bits.

g If the application client or device server detects a:

a) LOGICAL BLOCK APPLICATION TAG field set to FFFFh and type 1 protection (see SBC-3) or type 2 protection (see SBC-3) is enabled; or

b) LOGICAL BLOCK APPLICATION TAG field set to FFFFh, LOGICAL BLOCK REFERENCE TAG field set to FFFF FFFFh, and type 3 protection (see SBC-3) is enabled,

then the device server shall not check any protection information in the associated logical block.

h If type 1 protection is enabled, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the logical block reference tag if it has knowledge of the contents of the LOGICAL BLOCK REFERENCE TAG field. If type 2 protection is enabled, then this knowledge may be acquired through the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field in a VERIFY (32) command (see 3.60). If type 3 protection is enabled, then the method for acquiring this knowledge is not defined by this manual.

If the BYTCHK bit is set to one, the device server shall check the protection information read from the medium based on the VRPROTECT field as described in table 182.

Table 182. VRPROTECT field with BYTCHK set to one - medium (Sheet 1 of 2)

<table>
<tr><th>Code</th><th>Logical unit formatted with protection information</th><th>Field in protection information [g]</th><th>Extended INQUIRY Data VPD page bit value [f]</th><th>If check fails [d] [e], additional sense code</th></tr>
<tr><td rowspan="7">000b</td><td rowspan="6">Yes</td><td rowspan="2">LOGICAL BLOCK GUARD</td><td>GRD_CHK = 1</td><td>LOGICAL BLOCK GUARD CHECK FAILED</td></tr>
<tr><td>GRD_CHK = 0</td><td>No check performed</td></tr>
<tr><td rowspan="2">LOGICAL BLOCK APPLICATION TAG</td><td>APP_CHK = 1 [c] [g]</td><td>LOGICAL BLOCK APPLICATION TAG CHECK FAILED</td></tr>
<tr><td>APP_CHK = 0</td><td>No check performed</td></tr>
<tr><td rowspan="2">LOGICAL BLOCK REFERENCE TAG</td><td>REF_CHK = 1 [h]</td><td>LOGICAL BLOCK REFERENCE TAG CHECK FAILED</td></tr>
<tr><td>REF_CHK = 0</td><td>No check performed</td></tr>
<tr><td>No</td><td colspan="3">No protection information on the medium available to check</td></tr>
<tr><td rowspan="4">001b
010b
011b
100b
101b
[b]</td><td rowspan="3">Yes</td><td>LOGICAL BLOCK GUARD</td><td colspan="2">No check performed</td></tr>
<tr><td>LOGICAL BLOCK APPLICATION TAG</td><td colspan="2">No check performed</td></tr>
<tr><td>LOGICAL BLOCK REFERENCE TAG</td><td colspan="2">No check performed</td></tr>
<tr><td>No</td><td colspan="3">Error condition [a]</td></tr>
<tr><td>110b -
111b</td><td colspan="4">Reserved</td></tr>
</table>

Table 182. VRPROTECT field with BYTCHK set to one - medium (Sheet 2 of 2)

Code	Logical unit formatted with protection information	Field in protection information [g]	Extended INQUIRY Data VPD page bit value [f]	If check fails [d] [e], additional sense code

[a] A verify operation to a logical unit that supports protection information and has not been formatted with protection information 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.

[b] If the logical unit does not support protection information the requested command should 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.

[c] The device server shall check the logical block application tag if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the VERIFY (32) command (see 3.60) is used and the ATO bit is set to one in the Control mode page (see SPC-4), this knowledge is acquired from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge may be obtained by a method not defined by this manual.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] If multiple errors occur, the selection of which error to report is not defined by this manual.

[f] See the Extended INQUIRY Data VPD page (see SPC-4) for the definitions of the GRD_CHK bit, the APP_CHK bit, and the REF_CHK bit.

[g] If the application client or device server detects a:

a) logical block application tag field set to FFFFh and type 1 protection or type 2 protection is enabled; or
b) logical block application tag field set to FFFFh, logical block reference tag field set to FFFF FFFFh, and type 3 protection is enabled,

then the device server shall not check any protection information in the associated logical block.

[h] If type 1 protection is enabled, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the logical block reference tag if it has knowledge of the contents of the logical block reference tag field. If type 2 protection is enabled, then this knowledge may be acquired through the expected initial logical block reference tag field in a VERIFY (32) command (see 3.60). If type 3 protection is enabled, then the method for acquiring this knowledge is not defined by this manual.

If the BYTCHK bit is set to one, the device server shall check the protection information transferred from the data-out buffer based on the VRPROTECT field as described in table 183.

Table 183. VRPROTECT field with BYTCHK set to one - data-out buffer (Sheet 1 of 2)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d] [e], additional sense code
000b	Yes	No protection information received from application client to check		
	No	No protection information received from application client to check		
001b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	Shall [f]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	May [f]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
011b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		

Table 183. VRPROTECT field with BYTCHK set to one - data-out buffer (Sheet 2 of 2)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d] [e], additional sense code
100b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No	Error condition [a]		
101b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLI-CATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFER-ENCE TAG	May [f]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
110b - 111b	Reserved			

a A verify operation to a logical unit that supports protection information and has not been formatted with protection information 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.

b If the logical unit does not support protection information the requested command should 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.

c The device server may check the logical block application tag if the ATO bit is set to one in the Control mode page (see SPC-4) and if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the VERIFY (32) command (see 3.60) is used, this knowledge is obtained from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG field in the CDB. Otherwise, this knowledge is obtained by a method not defined by this manual.

d If an error is reported, the sense key shall be set to ABORTED COMMAND.

e If multiple errors occur, the selection of which error to report is not defined by this manual.

f If type 1 protection is enabled, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the logical block reference tag if it has knowledge of the contents of the logical block reference tag field. If type 2 protection is enabled, then this knowledge may be acquired through the expected initial logical block reference tag field in a VERIFY (32) command (see 3.60). If type 3 protection is enabled, then the method for acquiring this knowledge is not defined by this manual.

If the BYTCHK bit is set to one, the device server shall perform a byte-by-byte comparison of protection information transferred from the data-out buffer with protection information read from the medium based on the VRPROTECT field as described in table 184.

Table 184. VRPROTECT field with BYTCHK set to one - byte-by-byte comparison requirements (Sheet 1 of 4)

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails [c d], additional sense code
000b	Yes	No protection information received from application client to compare. Only user data is compared within each logical block.		
	No	No protection information or the medium or received from application client to compare. Only user data is compared within each logical block.		
001b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REFERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 0)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG (type 3 and ATO = 1)	Shall not	No compare performed
	No	Error condition [a]		

Table 184. VRPROTECT field with BYTCHK set to one - byte-by-byte comparison requirements (Sheet 2 of 4)

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails [c] [d], additional sense code
010b [b]	Yes	LOGICAL BLOCK GUARD	Shall not	No compare performed
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REF-ERENCE TAG (not type 3)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REF-ERENCE TAG (type 3 and ATO = 0	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REF-ERENCE TAG (type 3 and ATO = 1)	Shall not	No compare performed
	No	Error condition [a]		

Table 184. VRPROTECT field with BYTCHK set to one - byte-by-byte comparison requirements (Sheet 3 of 4)

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails [c d], additional sense code
010b [b] 100b	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [e]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) [f]	Shall not	No compare performed
		LOGICAL BLOCK REF-ERENCE TAG (not type 3)	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REF-ERENCE TAG (type 3 and ATO = 0	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
		LOGICAL BLOCK REF-ERENCE TAG (type 3 and ATO = 1)	Shall not	No compare performed
	No	Error condition [a]		
101b [b]	Yes	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 1) [E]	Shall	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG (ATO = 0) [F]	Shall not	No compare performed
		LOGICAL BLOCK REF-ERENCE TAG	Shall	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No	Error condition [a]		
110b - 111b	Reserved			

Table 184. VRPROTECT field with BYTCHK set to one - byte-by-byte comparison requirements (Sheet 4 of 4)

Code	Logical unit formatted with protection information	Field	Byte-by-byte Comparison	If compare fails [c d], additional sense code

a A verify operation to a logical unit that supports protection information and has not been formatted with protection information 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.
b If the logical unit does not support protection information the requested command should 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.
c If an error is reported, the sense key shall be set to MISCOMPARE.
d If multiple errors occur, the selection of which error to report is not defined by this manual.
e If the ATO bit is set to one in the Control mode page (see SPC-4), the logical block application tag shall not be modified by a device server.
f If the ATO bit is set to zero in the Control mode page (see SPC-4), the logical block application tag may be modified by a device server.

3.58 VERIFY (12) command

The VERIFY (12) command (see table 185) requests that the device server verify the specified logical block(s) on the medium. Each logical block includes user data and may include protection information, based on the VRPROTECT field and the medium format.

Table 185. VERIFY (12) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (AFh)							
1	VRPROTECT			DPO	Reserved		BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	(MSB)	VERIFICATION LENGTH						
9								(LSB)
10	RESTRICTED FOR MMC-5	Reserved		GROUP NUMBER				
11	CONTROL							

See the VERIFY (10) command (see 3.57) for the definitions of the fields in this command.

3.59 VERIFY (16) command

The VERIFY (16) command (see table 186) requests that the device server verify the specified logical block(s) on the medium. Each logical block includes user data and may include protection information, based on the VRPROTECT field and the medium format.

Table 186. VERIFY (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Fh)							
1	VRPROTECT			DPO	Reserved		BYTCHK	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	VERIFICATION LENGTH						
13								(LSB)
14	Restricted for MMC-5	Reserved		GROUP NUMBER				
15	CONTROL							

See the VERIFY (10) command (see 3.57) for the definitions of the fields in this command.

3.60 VERIFY (32) command

The VERIFY (32) command (see table 187) requests that the device server verify the specified logical block(s) on the medium. Each logical block includes user data and may include protection information, based on the VRPROTECT field and the medium format.

The VERIFY (32) command shall only be processed if type 2 protection is enabled (see SPC-4).

Table 187. VERIFY (32) command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
5								
6	Reserved			GROUP NUMBER				
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Ah)						
9								(LSB)
10	VRPROTECT			DPO	Reserved		BYTCHK	Reserved
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23								(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK						
27								(LSB)
28	(MSB)	VERIFICATION LENGTH						
31								(LSB)

See the VERIFY (10) command (see 3.57) for the definitions of the GROUP NUMBER field, VRPROTECT field, DPO bit, BYTCHK bit, LOGICAL BLOCK ADDRESS field, and VERIFICATION LENGTH field.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 181, table 182, table 183, and table 184 in 3.57), the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA.

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 181, table 182, table 183, and table 184 in 3.57), the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in the protection information.

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see table 181, table 182, table 183, and table 184 in 3.57), or if the ATO bit is set to zero, the LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored.

The LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored if:

a) the ATO bit is set to zero; or
b) the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see table 181, table 182, table 183, and table 184 in 3.57).

3.61 WRITE (6) command

The WRITE (6) command (see table 188) requests that the device server transfer the specified logical block(s) from the data-out buffer and write them. Each logical block transferred includes user data but does not include protection information. Each logical block written includes user data and, if the medium is formatted with protection information enabled, protection information.

Table 188. WRITE (6) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (0Ah)							
1	Reserved			(MSB)				
2	LOGICAL BLOCK ADDRESS							
3								(LSB)
4	TRANSFER LENGTH							
5	CONTROL							

The cache control bits are not provided for this command. Direct-access block devices with cache may have values for the cache control bits that may affect the WRITE (6) command, however no default value is defined by this manual. If explicit control is required, the WRITE (10) command should be used.

LOGICAL BLOCK ADDRESS field

See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field.

TRANSFER LENGTH field

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred from the data-out buffer and written, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A TRANSFER LENGTH field set to zero specifies that 256 logical blocks shall be written. Any other value specifies the number of logical blocks that shall be written. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page.

Note. For the WRITE (10) command, WRITE (12) command, WRITE (16) command, and WRITE (32) command, a TRANSFER LENGTH field set to zero specifies that no logical blocks are transferred.

If a WRITE (6) command is received after protection information is enabled the device server shall set the protection information as follows as it writes each logical block to the medium:

a) the LOGICAL BLOCK GUARD field set to a properly generated CRC (see SPC-4 4.16);
b) the LOGICAL BLOCK REFERENCE TAG field set to:
   A) the least significant four bytes of the LBA, if type 1 protection (see SPC-4) is enabled;

      or
   B) FFFFFFFFh, if type 2 protection or type 3 protection is enabled

      and
a) the LOGICAL BLOCK APPLICATION TAG field set to:
   A) FFFFh, if the ATO bit is set to one in the Control mode page (see 4.3.8); or
   B) any value, if the ATO bit is set to zero in the Control mode page (see 4.3.8).

3.62 WRITE (10) command

The WRITE (10) command (see table 189) requests that the device server transfer the specified logical block(s) from the data-out buffer and write them. Each logical block transferred includes user data and may include protection information, based on the WRPROTECT field and the medium format. Each logical block written includes user data and, if the medium is formatted with protection information enabled, protection information.

Table 189. WRITE (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Ah)							
1	WRPROTECT			DPO	FUA	Reserved	FUA_NV	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved			GROUP NUMBER				
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

See the READ (10) command (see 3.20) for the definition of the DPO bit. See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field. See 2.2.8 and SPC-4 for the definition of the GROUP NUMBER field.

The device server shall check the protection information transferred from the data-out buffer based on the WRPROTECT field as described in table 190.

Table 190. WRPROTECT field (Sheet 1 of 4)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d i], additional sense code
000b	Yes [f g h]	No protection information received from application client to check		
	No	No protection information received from application client to check		

Table 190. WRPROTECT field (Sheet 2 of 4)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d i], additional sense code
001b [b]	**Yes [e]**	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	Shall (except for type 3)[j]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No [a]	No protection information available to check		
010b [b]	**Yes [e]**	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	May [k]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No [a]	No protection information available to check		
011b [b]	**Yes [e]**	LOGICAL BLOCK GUARD	Shall not	No check performed
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No [a]	No protection information available to check		

Table 190. WRPROTECT field (Sheet 3 of 4)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d] [i], additional sense code
100b [b]	Yes [e]	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	Shall not	No check performed
		LOGICAL BLOCK REFERENCE TAG	Shall not	No check performed
	No [a]	No protection information available to check		
101b [b]	Yes [e]	LOGICAL BLOCK GUARD	Shall	LOGICAL BLOCK GUARD CHECK FAILED
		LOGICAL BLOCK APPLICATION TAG	May [c]	LOGICAL BLOCK APPLICATION TAG CHECK FAILED
		LOGICAL BLOCK REFERENCE TAG	May [j]	LOGICAL BLOCK REFERENCE TAG CHECK FAILED
	No [a]	No protection information available to check		

Table 190. WRPROTECT field (Sheet 4 of 4)

Code	Logical unit formatted with protection information	Field in protection information	Device server check	If check fails [d i], additional sense code
110b - 111b	Reserved			

[a] A write operation to a logical unit that supports protection information and has not been formatted with protection information 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.

[b] If the logical unit does not support protection information the requested command should 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.

[c] The device server may check the logical block application tag if the ATO bit is set to one in the Control mode page (see SPC-4) and if it has knowledge of the contents of the LOGICAL BLOCK APPLICATION TAG field. If the WRITE (32) command (see 3.65) is used, this knowledge is obtained from the EXPECTED LOGICAL BLOCK APPLICATION TAG field and the LOGICAL BLOCK APPLICATION TAG MASK field in the CDB. Otherwise, this knowledge is obtained by a method not defined by this manual.

[d] If an error is reported, the sense key shall be set to ABORTED COMMAND.

[e] Device server shall preserve the contents of protection information (e.g., write to medium, store in non-volatile memory).

[f] The device server shall write a properly generated CRC into each LOGICAL BLOCK GUARD field.

[g] If type 1 protection is enabled (see SBC-3), the device server shall write the least significant four bytes of each LBA into the LOGICAL BLOCK REFERENCE TAG field of each of the written logical blocks. If the P_TYPE field is not set to 000b, the device server shall write a value of FFFFFFFFh into the logical block reference tag field of each of the written logical blocks.

[h] If the ATO bit is set to one in the Control mode page (see 4.3.8), the device server shall write FFFFh into each LOGICAL BLOCK APPLICATION TAG field. If the ATO bit is set to zero, the device server may write any value into each LOGICAL BLOCK APPLICATION TAG field.

[i] If multiple errors occur, the selection of which error to report is not defined by this manual.

[j] If type 1 protection is enabled, the device server checks the logical block reference tag by comparing it to the lower 4 bytes of the LBA associated with the logical block. If type 2 protection or type 3 protection is enabled, the device server checks the logical block reference tag if it has knowledge of the contents of the logical block reference tag field. If type 2 protection is enabled, then this knowledge may be acquired through the expected initial logical block reference tag field in a WRITE (32) command (see 3.65). If type 3 protection is enabled, then the method for acquiring this knowledge is not defined by this manual.

The force unit access (FUA) and force unit access non-volatile cache (FUA_NV) bits are defined in table 191.

Table 191. Force unit access for write operations

FUA	FUA_NV	Description
0	0	The device server shall write the logical blocks to volatile cache, non-volatile cache, and/or the medium.
0	1	If the NV_SUP bit is set to one in the Extended INQUIRY Data VPD page (see SPC-4), the device server shall write the logical blocks to non-volatile cache and/or the medium. If the NV_SUP bit is set to zero in the Extended INQUIRY Data VPD page (see SPC-4), the device server shall write the logical blocks to volatile cache, non-volatile cache, and/or the medium.
1	0 or 1	The device server shall write the logical blocks to the medium, and shall not return GOOD status until the logical blocks have actually been written on the medium.

If logical blocks are transferred directly to a cache, the device server may return GOOD status prior to writing the logical blocks to the medium. Any error that occurs after the GOOD status is returned is a deferred error, and information regarding the error is not reported until a subsequent command.

TRANSFER LENGTH field

The TRANSFER LENGTH field specifies the number of contiguous logical blocks of data that shall be transferred from the data-out buffer and written, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A TRANSFER LENGTH field set to zero specifies that no logical blocks shall be written. This condition shall not be considered an error. Any other value specifies the number of logical blocks that shall be written. If the logical block address plus the transfer length exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE. The TRANSFER LENGTH field is constrained by the MAXIMUM TRANSFER LENGTH field in the Block Limits VPD page.

Note. For the WRITE (6) command, a TRANSFER LENGTH field set to zero specifies that 256 logical blocks are transferred.

3.63 WRITE (12) command

The WRITE (12) command (see table 192) requests that the device server transfer the specified logical block(s) from the data-out buffer and write them. Each logical block transferred includes user data and may include protection information, based on the WRPROTECT field and the medium format. Each logical block written includes user data and, if the medium is formatted with protection information enabled, protection information.

Table 192. WRITE (12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (AAh)							
1	WRPROTECT			DPO	FUA	Reserved	FUA_NV	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	Restricted for MMC-5	Reserved		GROUP NUMBER				
11	CONTROL							

See the WRITE (10) command (see 3.62) for the definitions of the fields in this command.

3.64 WRITE (16) command

The WRITE (16) command (see table 193) requests that the device server transfer the specified logical block(s) from the data-out buffer and write them. Each logical block transferred includes user data and may include protection information, based on the WRPROTECT field and the medium format. Each logical block written includes user data and, if the medium is formatted with protection information enabled, protection information.

Table 193. WRITE (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Ah)							
1	WRPROTECT			DPO	FUA	Reserved	FUA_NV	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	TRANSFER LENGTH						
13								(LSB)
14	Restricted for MMC-5	Reserved		GROUP NUMBER				
15	CONTROL							

See the WRITE (10) command (see 3.62) for the definitions of the fields in this command.

3.65 WRITE (32) command

The WRITE (32) command (see table 194) requests that the device server transfer the specified logical block(s) from the data-out buffer and write them. Each logical block transferred includes user data and may include protection information, based on the WRPROTECT field and the medium format. Each logical block written includes user data and, if the medium is formatted with protection information enabled, protection information.

The WRITE (32) command shall only be processed if type 2 protection is enabled (see SBC-3).

Table 194. WRITE (32) command

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td colspan="8">OPERATION CODE (7Fh)</td></tr>
<tr><td>1</td><td colspan="8">CONTROL</td></tr>
<tr><td>2
5</td><td colspan="8">Reserved</td></tr>
<tr><td>6</td><td colspan="3">Reserved</td><td colspan="5">GROUP NUMBER</td></tr>
<tr><td>7</td><td colspan="8">ADDITIONAL CDB LENGTH (18h)</td></tr>
<tr><td>8
9</td><td colspan="8">(MSB)
SERVICE ACTION (000Bh)
(LSB)</td></tr>
<tr><td>10</td><td colspan="3">WRPROTECT</td><td>DPO</td><td>FUA</td><td>Reserved</td><td>FUA_NV</td><td>Reserved</td></tr>
<tr><td>11</td><td colspan="8">Reserved</td></tr>
<tr><td>12
19</td><td colspan="8">(MSB)
LOGICAL BLOCK ADDRESS
(LSB)</td></tr>
<tr><td>20
23</td><td colspan="8">(MSB)
EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG
(LSB)</td></tr>
<tr><td>24
25</td><td colspan="8">(MSB)
EXPECTED LOGICAL BLOCK APPLICATION TAG
(LSB)</td></tr>
<tr><td>26
27</td><td colspan="8">(MSB)
LOGICAL BLOCK APPLICATION TAG MASK
(LSB)</td></tr>
<tr><td>28
31</td><td colspan="8">(MSB)
TRANSFER LENGTH
(LSB)</td></tr>
</table>

See the WRITE (10) command (see 3.62) for the definitions of the GROUP NUMBER field, the WRPROTECT field, the DPO bit, the FUA bit, the FUA_NV bit, the LOGICAL BLOCK ADDRESS field, and the TRANSFER LENGTH field.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 190 in 3.62), the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA.

If the ATO bit is set to one in the Control mode page (see SPC-4) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 190 in 3.62), the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in the protection information.

The LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored if:

a) the ATO bit is set to zero; or
b) the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see SBC-3).

3.66 WRITE AND VERIFY (10) command

The WRITE AND VERIFY (10) command (see table 195) requests that the device server transfer the specified logical block(s) from the data-out buffer, write them to the medium, and then verify that they are correctly written. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format. The logical blocks are only transferred once from the data-out buffer to the device server.

Table 195. WRITE AND VERIFY (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (2Eh)							
1	WRPROTECT			DPO	Reserved		BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved			GROUP NUMBER				
7	(MSB)	TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field. See 2.2.3 for the definition of the GROUP NUMBER field. See the WRITE (10) command (see 3.62) for the definitions of the TRANSFER LENGTH field and the WRPROTECT field. See the READ (10) command (see 3.20) for the definition of the DPO bit.

If the Verify Error Recovery mode page is also implemented, then the current settings in that mode page along with the AWRE bit in the Read-Write Error Recovery mode page specify the verification error criteria. If these mode pages are not implemented, then the verification criteria is vendor-specific.

BYTCHK bit

0 A byte check (BYTCHK) bit set to zero specifies that, after writing, the device server perform a medium verification with no data comparison.

1 A BYTCHK bit set to one specifies that, after writing, the device server perform a byte-by-byte comparison of data written on the medium with the data just written. If the comparison is unsuccessful for any reason, the device server shall terminate the command with CHECK CONDITION status with the sense key set to MISCOMPARE and the additional sense code set to the appropriate value for the condition.

3.67 WRITE AND VERIFY (12) command

The WRITE AND VERIFY (12) command (see table 196) requests that the device server transfer the specified logical block(s) from the data-out buffer, write them to the medium, and then verify that they are correctly written. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format. The logical blocks are only transferred once from the data-out buffer to the device server.

Table 196. WRITE AND VERIFY (12) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (AEh)							
1	WRPROTECT			DPO	Reserved		BYTCHK	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	(MSB)	TRANSFER LENGTH						
9								(LSB)
10	Restricted for MMC-5	Reserved		GROUP NUMBER				
11	CONTROL							

See the WRITE AND VERIFY (10) command (see 3.66) for the definitions of the fields in this command.

3.68 WRITE AND VERIFY (16) command

The WRITE AND VERIFY (16) command (see table 197) requests that the device server transfer the specified logical block(s) from the data-out buffer, write them to the medium, and then verify that they are correctly written. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format. The logical blocks are only transferred once from the data-out buffer to the device server.

Table 197. WRITE AND VERIFY (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (8Eh)							
1	WRPROTECT			DPO	Reserved		BYTCHK	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	TRANSFER LENGTH						
13								(LSB)
14	Restricted for MMC-5	Reserved		GROUP NUMBER				
15	CONTROL							

See the WRITE AND VERIFY (10) command (see 3.66) for the definitions of the fields in this command.

3.69 WRITE AND VERIFY (32) command

The WRITE AND VERIFY (32) command (see table 198) requests that the device server transfer the specified logical block(s) from the data-out buffer, write them to the medium, and then verify that they are correctly written. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format. The logical blocks are only transferred once from the data-out buffer to the device server.

Table 198. WRITE AND VERIFY (32) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2 – 5	Reserved							
6	Reserved			GROUP NUMBER				
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Ch)						
9								(LSB)
10	WRPROTECT			DPO	Reserved		BYTCHK	Reserved
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23								(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK						
27								(LSB)
28	(MSB)	TRANSFER LENGTH						
31								(LSB)

See the WRITE AND VERIFY (10) command (see 3.66) for the definitions of the GROUP NUMBER field, the WRPROTECT field, the DPO bit, the BYTCHK bit, the LOGICAL BLOCK ADDRESS field, and the TRANSFER LENGTH field.

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see table 190), the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA.

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see table 190), the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in the protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in the protection information.

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see table 190), or if the ATO bit is set to zero, the LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored.

3.70 WRITE BUFFER command

3.70.1 WRITE BUFFER command introduction

The WRITE BUFFER command (see table 199) is used in conjunction with the READ BUFFER command for:

a) testing logical unit buffer memory;
b) testing the integrity of the service delivery subsystem;
c) downloading microcode; and
d) downloading application client error history (see SPC-4).

Table 199. WRITE BUFFER command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Bh)							
1	Reserved			MODE				
2	BUFFER ID							
3	(MSB)	BUFFER OFFSET						
5								(LSB)
6	(MSB)	PARAMETER LIST LENGTH						
8								(LSB)
9	CONTROL							

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

The function of this command and the meaning of fields within the CDB depend on the contents of the MODE field. The MODE field is defined in table 200.

Table 200. WRITE BUFFER MODE field (Sheet 1 of 2)

Mode	Description
00h	Combined header and data [a]
01h	Not supported
02h	Data
04h	Download microcode and activate
05h	Download microcode, save and activate

a Mode 00h is not recommended.
b When downloading microcode with buffer offsets, the WRITE BUFFER command mode should be 06h or 07h.

Table 200. WRITE BUFFER MODE field (Sheet 2 of 2)

Mode	Description
06h	Download microcode with offsets [b] and activate
07h	Download microcode with offsets [b], save, and activate
08h - 09h	Reserved
0Ah	Echo buffer
0Bh - 0Dh	Reserved
0Eh	Download microcode with offsets, save, and defer active
0Fh	Activate deferred microcode
10h - 19h	Reserved
1Ah	Enable expander communications protocol and Echo buffer
1Bh	Disable expander communications protocol
1Ch	Download application log
1Dh - 1Fh	Reserved

a Mode 00h is not recommended.
b When downloading microcode with buffer offsets, the WRITE BUFFER command mode should be 06h or 07h.

3.70.2 Combined header and data mode (00h)

Note. This mode is not recommended.

In this mode, data to be transferred is preceded by a four-byte header. The four-byte header consists of all reserved bytes. The BUFFER ID and the BUFFER OFFSET fields shall be zero.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer. This number includes four bytes of header, so the data length to be stored in the device server's buffer is parameter list length minus four. The application client should attempt to ensure that the parameter list length is not greater than four plus the BUFFER CAPACITY field value (see table 99) that is returned in the header of the READ BUFFER command (mode 0h). If the parameter list length exceeds the buffer capacity, the 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.

3.70.3 Vendor specific mode (01h)

This mode is not supported by Seagate products. If used, the drive will return CHECK CONDITION and report an ILLEGAL REQUEST/INVALID FIELD IN CDB error.

3.70.4 Data mode (02h)

In this mode, the Data-Out Buffer contains buffer data destined for the logical unit. The BUFFER ID field identifies a specific buffer within the logical unit. Seagate assigns buffer ID codes to buffers within the logical unit. Buffer ID zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is selected, the 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.

Data are written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor. If the device server is unable to accept the specified buffer offset, the 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.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should attempt to ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see table 99). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, the 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.

3.70.5 Download microcode and activate mode (04h)

In this mode, vendor specific microcode or control information shall be transferred to the control memory space of the logical unit. After a hard reset, the device operation shall revert to a vendor specific condition. The meanings of the BUFFER ID, BUFFER OFFSET, and PARAMETER LIST LENGTH fields are not specified by this manual and are not required to be zero-filled. When the microcode download has completed successfully the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus on which the WRITE BUFFER command was received, with the additional sense code set to MICROCODE HAS BEEN CHANGED.

If the logical unit is unable to accept this command because of some device condition, each WRITE BUFFER command with this mode (04h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

The downloaded microcode may or may not be activated after the WRITE BUFFER command completes and shall be activated when one of the following occurs:

a) power on; or
b) after each hard reset.

3.70.6 Download microcode, save, and activate mode (05h)

In this mode, vendor specific microcode or control information shall be transferred to the logical unit and, if the WRITE BUFFER command is completed successfully, also shall be saved in a nonvolatile memory space (e.g., semiconductor, disk, or other). The downloaded code shall then be effective after each hard reset until it is supplanted in another download microcode and save operation or download microcode with offsets and save operation. The meanings of the BUFFER ID, BUFFER OFFSET, and PARAMETER LIST LENGTH fields are not specified by this manual and are not required to be zero-filled. When the download microcode and save command has completed successfully the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus on which the WRITE BUFFER command was received with the additional sense code set to MICROCODE HAS BEEN CHANGED.

If the logical unit is unable to accept this command because of some device condition, each WRITE BUFFER command with this mode (05h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

The downloaded microcode may or may not be activated after the WRITE BUFFER command completes and shall be activated when one of the following occurs:

a) power on; or
b) after each hard reset.

3.70.7 Download microcode with offsets and activate mode (06h)

In this mode, the application client may split the transfer of Seagate specific microcode or control information over two or more WRITE BUFFER commands. If the last WRITE BUFFER command of a set of one or more commands completes successfully, then the microcode or control information shall be transferred to the control memory space of the logical unit. After a hard reset, the device shall revert to a vendor specific condition. In this mode, the Data-Out Buffer contains vendor specific, self-describing microcode or control information.

Since the downloaded microcode or control information may be sent using several commands, when the logical unit detects the last download microcode with offsets WRITE BUFFER command has been received, the device server shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command. After the last command completes successfully the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus 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.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (i.e., one or more commands) are not received before a logical unit reset or I_T nexus loss occurs, the change shall not be effective and the new microcode or control information shall be discarded.

The BUFFER ID field specifies a buffer within the logical unit. Seagate assigns buffer ID codes to buffers within the logical unit. A buffer ID value of zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is specified, the 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.

The microcode or control information are written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field. The application client shall send commands that conform to the offset boundary requirements. If the device server is unable to accept the specified buffer offset, the 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.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see table 99). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, then the 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.

If the logical unit is unable to accept this command because of some device condition, each WRITE BUFFER command with this mode (06h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

3.70.8 Download microcode with offsets and save mode (07h)

In this mode, the application client may split the transfer of Seagate specific microcode or control information over two or more WRITE BUFFER commands. If the last WRITE BUFFER command of a set of one or more commands completes successfully, then the microcode or control information shall be saved in a nonvolatile memory space (e.g., semiconductor, disk, or other). The saved downloaded microcode or control information shall then be effective after each hard reset until it is supplanted by another download microcode with save operation or download microcode with offsets and save operation. In this mode, the Data-Out Buffer contains vendor specific, self-describing microcode or control information.

Since the downloaded microcode or control information may be sent using several commands, when the logical unit detects the last download microcode with offsets and save mode WRITE BUFFER command has been received, the device server shall perform any logical unit required verification of the complete set of downloaded microcode or control information prior to returning GOOD status for the last command. After the last command completes successfully the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus 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.

If the complete set of WRITE BUFFER commands required to effect a microcode or control information change (i.e., one or more commands) are not received before a logical unit reset or I_T nexus loss occurs, the change shall not be effective and the new microcode or control information shall be discarded.

The BUFFER ID field specifies a buffer within the logical unit. Seagate assigns buffer ID codes to buffers within the logical unit. A buffer ID value of zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. If an unsupported buffer ID code is specified, the 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.

The microcode or control information are written to the logical unit buffer starting at the location specified by the BUFFER OFFSET field. The application client shall conform to the offset boundary requirements. If the device server is unable to accept the specified buffer offset, the 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.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be present in the Data-Out Buffer to be stored in the specified buffer beginning at the buffer offset. The application client should ensure that the parameter list length plus the buffer offset does not exceed the capacity of the specified buffer. The capacity of the buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER descriptor (see Table 99.). If the BUFFER OFFSET and PARAMETER LIST LENGTH fields specify a transfer in excess of the buffer capacity, then the 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.

If the logical unit is unable to accept this command because of some device condition, each WRITE BUFFER command with this mode (07h) shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

The downloaded microcode may or may not be activated after the WRITE BUFFER command completes and shall be activated when one of the following occurs:

a) power on; or
b) after each hard reset.

3.70.9 Write data to echo buffer mode (0Ah)

In this mode the device server transfers data from the application client and stores it in an echo buffer. An echo buffer is assigned in the same manner by the device server as it would for a write operation. Data shall be sent aligned on four-byte boundaries. The BUFFER ID and BUFFER OFFSET fields are ignored in this mode.

Note. It is recommended that the logical unit assign echo buffers on a per I_T nexus basis to limit the number of exception conditions that may occur when I_T nexuses are present.

Upon successful completion of a WRITE BUFFER command the data shall be preserved in the echo buffer unless there is an intervening command to any logical unit in which case the data may be changed.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the echo buffer. The application client should ensure that the parameter list length does not exceed the capacity of the echo buffer. The capacity of the echo buffer is indicated by the BUFFER CAPACITY field in the READ BUFFER echo buffer descriptor (see Table 101.). If the PARAMETER LIST LENGTH field specifies a transfer in excess of the buffer capacity, the 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.

3.70.10 Download microcode with offsets, save, and defer activate mode (0Eh)

In this mode, microcode shall be transferred to the device server using one or more WRITE BUFFER commands, saved to nonvolatile storage, and considered deferred (see SPC-4).

The deferred microcode shall be activated and no longer considered deferred when one of the following occurs:

a) a power on;
b) a hard reset;
c) a START STOP UNIT command is processed (see 3.52);
d) a FORMAT UNIT command is processed (see 3.5); or
e) a WRITE BUFFER command with the activate deferred microcode mode (0Fh) is processed (see 3.70.11).

The BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field are defined in the download microcode with offsets mode (see 3.70.7).

3.70.11 Activate deferred microcode mode (0Fh)

In this mode, deferred microcode that has been saved using the download microcode with offsets, save, and defer activate mode (see 3.70.10), if any, shall be activated and no longer considered deferred (see SPC-4).

The the BUFFER ID field, BUFFER OFFSET field, and PARAMETER LIST LENGTH field shall be ignored in this mode.

If there is no deferred microcode that has been saved using the download microcode with offsets, save, and defer activate mode, then the WRITE BUFFER command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

3.70.12 Enable expander communications protocol and Echo buffer mode (1Ah)

Receipt of a WRITE BUFFER command with this mode (1Ah) causes a communicative expander (see SPI-5) to enter the expanded communications protocol mode. Device servers in SCSI target devices that receive a WRITE BUFFER command with this mode shall process it as if it were a WRITE BUFFER command with mode 0Ah (see 3.70.9).

3.70.13 Disable expander communications protocol mode (1Bh)

Receipt of a WRITE BUFFER command with this mode (1Bh) causes a communicative expander (see SPI-5) to exit the expanded communications protocol mode and return to simple expander operation. Device servers in SCSI target devices that receive a WRITE BUFFER command with this mode 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.

3.70.14 Download application log mode (1Ch)

In this mode the device server transfers data from the application client and stores it in an application log. The format of the application log data is as specified in table 201.

The BUFFER ID field and BUFFER OFFSET field are ignored in this mode.

Upon successful completion of a WRITE BUFFER command, the information contained in the application client error history parameter list shall be appended to the application client error history in a format determined by the logical unit.

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the application log. If the PARAMETER LIST LENGTH field specifies a transfer that exceeds the application log's capacity, the 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

The device server shall not return an error based on the contents of any of the field values defined in table 201 except:

a) the CLR bit;
b) the ERROR LOCATION LENGTH field; and
c) the APPLICATION CLIENT ERROR HISTORY LENGTH field.

Table 201. Application log data WRITE BUFFER format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	T10 VENDOR IDENTIFICATION						
7								(LSB)
8	(MSB)	ERROR TYPE						
9								(LSB)
10	Reserved							
11								
12	(MSB)	TIME STAMP						
17								(LSB)
18	Reserved							
19								
20	Reserved				CODE SET			
21	ERROR LOCATION FORMAT							
22	(MSB)	ERROR LOCATION LENGTH (M-25)						
23								(LSB)
24	(MSB)	VENDOR SPECIFIC LENGTH (N-M)						
25								(LSB)
26	(MSB)	ERROR LOCATION						
m								(LSB)
m+1	Vendor specific							
n								

T10 VENDOR IDENTIFICATION field

The T10 VENDOR IDENTIFICATION field contains eight bytes of left-aligned ASCII data identifying the vendor of the product. The T10 vendor identification shall be one assigned by INCITS. A list of assigned T10 vendor identifications is in SPC-4 Annex E and on the T10 web site (http://www.T10.org).

ERROR TYPE field

The ERROR TYPE field (see table 202) specifies the error detected by the application client.

Table 202. ERROR TYPE field

Code	Description
0000h	No error specified by the application client
0001h	An unknown error was detected by the application client
0002h	The application client detected corrupted data
0003h	The application client detected a permanent error
0004h	The application client detected a service response of SERVICE DELIVERY OR TARGET FAILURE (SAM-4).
0005h - 7FFFh	Reserved
8000h - FFFFh	Vendor specific

If the CLR_SUP bit is set to one in the error history directory parameter data (see 3.70.14), a CLR bit set to one specifies that the device server shall:

a) clear the portions of the error history that the device server allows to be cleared; and
b) ignore any application client error history specified in the parameter list.

If the CLR_SUP bit is set to one in the error history directory parameter data, a CLR bit set to zero specifies that the device server shall not ignore the CLR bit.

If the CLR_SUP bit is set to one in the error history directory parameter data, a CLR bit set to zero specifies that the device server shall:

a) not clear the error history; and
b) process all application client error history specified in the parameter list.

If the CLR_SUP bit is set to zero in the error history directory parameter data, the device server shall ignore the CLR bit.

TIME STAMP field

The TIME STAMP field shall contain:

a) a time based on the timestamp reported by the REPORT TIMESTAMP command, if the device server supports a device clock (see SPC-4)

b) The number of milliseconds that have elapsed since midnight, 1 January 1970 UT; or

c) Zero, if the application client is not able to determine the UT of the log entry.

CODE SET field

The CODE SET field specifies the code set used for the application log information (see table 203) and shall only apply to information contained in the VENDOR SPECIFIC field.

Note. The CODE SET field is intended to be an aid to software that displays the application log information.

Table 203. CODE SET field

Code	Description
0h	Reserved
1h	The application log information is binary
2h	The application log information is ASCII printable characters (i.e., code values 20h through 7Eh)
3h	The application log information is ISO/IEC 10646-1 (UTF-8) codes
4h - Fh	Reserved

ERROR LOCATION FORMAT field

The ERROR LOCATION FORMAT field specifies the format (see table 204) of the ERROR LOCATION field.

Table 204. ERROR LOCATION FORMAT field

Code	Description
00h	No error specified by the application client
01h	The ERROR LOCATION field specifies the logical block (e.g., LBA) associated with the error information contained within the application log.
02h - 7Fh	Reserved
80h - FFh	Vendor specific

ERROR LOCATION LENGTH field

The ERROR LOCATION LENGTH field specifies the length of the ERROR LOCATION field. The ERROR LOCATION LENGTH field value shall be a multiple of four. An error location length value of zero specifies there is no error location information.

VENDOR SPECIFIC LENGTH field

The VENDOR SPECIFIC LENGTH field specifies the length of the VENDOR SPECIFIC field. The VENDOR SPECIFIC LENGTH field value shall be a multiple of four. A vendor specific length value of zero specifies there is no vendor specific information.

ERROR LOCATION field

The ERROR LOCATION field specifies the location at which the application client detected the error.

VENDOR SPECIFIC field

The VENDOR SPECIFIC field provides vendor specific information on the error.

3.71 WRITE LONG (10) command

The WRITE LONG (10) command (see table 205) requests that the device server transfer data for a single logical block from the data-out buffer and write it to the medium. The data written shall be the same length and shall be in the same order as the data returned by the READ LONG (10) command (see 3.29). The device server shall write the logical block to the medium, and shall not return GOOD status until the logical block has actually been written on the medium.

Table 205. WRITE LONG (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Fh)							
1	COR_DIS	WR_UNCOR	PBLOCK	Reserved				Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved							
7	(MSB)	BYTE TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

The OPERATION CODE field is defined in 2.2.1 shall be set to the value defined in table 205.

COR_DIS bit, WR_UNCOR bit and PBLOCK bit

The correction disabled (COR_DIS) bit, write uncorrectable error (WR_UNCOR) bit and physical block (PBLOCK) bit are define in table 206. If there are more than one logical block per physical block (i.e., the LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field in the READ CAPACITY (16) data (see 3.26) is set to a non-zero value), the device server shall support the WR_UNCOR bit and the PBLOCK bit.

Table 206. COR_DIS bit, WR_UNCOR bit and PBLOCK bit

COR_DIS	WR_UNCOR	PBLOCK	More than one logical block per physical block [a]	Description
0	0	0	yes or no	Write only the specified logical block using the value in the BYTE TRANSFER LENGTH field.
0	0	1	no	Terminate the WRITE LONG command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
0	0	1	yes	Write the entire physical block containing the specified logical block using the value in the BYTE TRANSFER LENGTH field.
0	1	0	yes or no	Mark only the specified logical block as containing a pseudo unrecovered error with correction enabled (see SBC-3) in a manner that causes the device server to perform the maximum error recovery as defined by the Read-Write Error Recovery mode page (see 4.3.15). Ignore the BYTE TRANSFER LENGTH field, and transfer no data.
0	1	1	no	Terminate the WRITE LONG command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.
0	1	1	yes	Mark the entire physical block containing the specified logical block as containing a pseudo unrecovered error with correction enabled (i.e., mark all of the logical blocks in the same physical block that contains the specified logical block as containing a pseudo unrecovered error with correction enabled) (see SBC-3). Ignore the BYTE TRANSFER LENGTH field, and transfer no data.

[a] An entry of “yes” means that the LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field in the READ CAPACITY (16) data (see 3.26) is set to a non-zero value. An entry of “no” means that the field is set to zero.

<table>
<tr><th>COR_DIS</th><th>WR_UNCOR</th><th>PBLOCK</th><th>More than one logical block per physical block [a]</th><th>Description</th></tr>
<tr><td rowspan="3">1</td><td rowspan="3">0</td><td>0</td><td>yes or no</td><td>Mark only the specified logical block as containing a pseudo unrecovered error with correction disabled (see SBC-3)

Write only the specified logical block using the value in the BYTE TRANSFER LENGTH field.</td></tr>
<tr><td rowspan="2">1</td><td>no</td><td>Terminate the WRITE LONG command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</td></tr>
<tr><td>yes</td><td>Mark the entire physical block containing the specified logical block as containing a pseudo unrecovered error with correction disabled (i.e., mark all of the logical blocks in the same physical block that contains the specified logical block as containing a pseudo unrecovered error with correction disabled) (see SBC-3).

Write the entire physical block containing the specified logical block using the value in the BYTE TRANSFER LENGTH field.</td></tr>
<tr><td rowspan="3">1</td><td rowspan="3">1</td><td>0</td><td>yes or no</td><td>Mark only the specified logical block as containing a pseudo unrecovered error with correction disabled (see SBC-3).

Ignore the BYTE TRANSFER LENGTH field, and transfer no data.</td></tr>
<tr><td rowspan="2">1</td><td>no</td><td>Terminate the WRITE LONG command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</td></tr>
<tr><td>yes</td><td>Mark the entire physical block containing the specified logical block as containing a pseudo unrecovered error with correction disabled (i.e., mark all of the logical blocks in the same physical block that contains the specified logical block as containing a pseudo unrecovered error with correction disabled) (see SBC-3).

Ignore the BYTE TRANSFER LENGTH field, and transfer no data.</td></tr>
<tr><td colspan="5">[a] An entry of “yes” means that the LOGICAL BLOCKS PER PHYSICAL BLOCK EXPONENT field in the READ CAPACITY (16) data (see 3.26) is set to a non-zero value. An entry of “no” means that the field is set to zero.</td></tr>
</table>

Any pseudo unrecovered error with correction disabled shall remain in effect until the logical block is written by any means (e.g., another WRITE LONG command with the COR_DIS bit set to zero and the WR_UNCOR bit set to zero that writes to the same logical block, any WRITE command that specifies the writes to the same logical block, or a FORMAT UNIT command).

In the Extended INQUIRY Data VPD page (see 4.4.6), the setting of the CD_SUP bit indicates whether or not the logical unit supports the CD_SUP bit being set to one, and the setting of the WU_SUP bit indicates whether or not the logical unit supports the WR_UNCOR bit being set to one.

LOGICAL BLOCK ADDRESS field

The LOGICAL BLOCK ADDRESS field (see 2.2.3) specifies an LBA. If the specified LBA exceeds the capacity of the medium, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE

BYTE TRANSFER LENGTH field

If table 206 defines that the value in the BYTE TRANSFER LENGTH field is used, then the BYTE TRANSFER LENGTH field specifies the number of bytes of data that the device server shall transfer from the data-out buffer and write to the specified logical block or physical block. If the BYTE TRANSFER LENGTH field is not set to zero and does not match the data length that the device server returns for a READ LONG command, then the device server 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. In the sense data (see 2.4), the ILI and VALID bits shall be set to one and the INFORMATION field shall be set to the difference (i.e., residue) of the requested length minus the actual length in bytes. Negative values shall be indicated by two's complement notation. If the BYTE TRANSFER LENGTH field is set to zero, then no bytes shall be written. This condition shall not be considered an error.

3.72 WRITE LONG (16) command

The WRITE LONG (16) command (see table 207) requests that the device server transfer data for a single logical block from the data-out buffer and write it to the medium. The data written shall be the same length and shall be in the same order as the data returned by the READ LONG (16) command (see 3.30). The device server shall write the logical block to the medium, and shall not return GOOD status until the logical block has actually been written on the medium. This command is implemented as a service action of the SERVICE ACTION OUT operation code.

Table 207. WRITE LONG (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Fh)							
1	COR_DIS	WR_UNCOR	PBLOCK	SERVICE ACTION (11h)				
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	Reserved							
11								
12	(MSB)	BYTE TRANSFER LENGTH						
13								(LSB)
14	Reserved							
15	CONTROL							

The OPERATION CODE field (see 2.2.1) and SERVICE ACTION field are defined in (see 2.2.2) and shall be set to the values defined in table 207.

See the WRITE LONG (10) command (see 3.71) for the definitions of the fields in this command.

3.73 WRITE SAME (10) command

The WRITE SAME (10) command (see table 208) requests that the device server transfer a single logical block from the data-out buffer and write the contents of that logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format.

Table 208. WRITE SAME (10) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (41h)							
1	WRPROTECT			Reserved		PBDATA	LBDATA	Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved			GROUP NUMBER				
7	(MSB)	NUMBER OF BLOCKS						
8								(LSB)
9	CONTROL							

WRPROTECT field

See the WRITE (10) command (see 3.62) for the definitions of the WRPROTECT field.

LOGICAL BLOCK ADDRESS field

See 2.2.3 for the definition of the LOGICAL BLOCK ADDRESS field.

GROUP NUMBER field

See clause 2.2.8 for the definition of the GROUP NUMBER field.

Table 209 describes the LBDATA bit and the PBDATA bit.

Table 209. LBDATA bit and PBDATA bit

LBDATA	PBDATA	Description
0	0	The device server shall write the single block of user data received from the data-out buffer to each logical block without modification. If the medium is formatted with type 1 or type 2 protection information (see SBC-2): **a** The device server shall place the value from each LOGICAL BLOCK REFERENCE TAG field received in the single block of data from the data-out buffer into the corresponding LOGICAL BLOCK REFERENCE TAG field of the first logical block written to the medium. The device sever shall place the value of the previous LOGICAL BLOCK REFERENCE TAG field plus one into each of the subsequent LOGICAL BLOCK REFERENCE TAG fields; **b** If the ATO bit is set to one in the Control mode page (see 4.3.8), then the device server shall place each logical block application tag received in the single block of data into the corresponding LOGICAL BLOCK APPLICATION TAG field of each logical block. If the ATO bit is set to zero, then the device server may write any value into the LOGICAL BLOCK APPLICATION TAG field(s) of each logical block; and **c** The device server shall place the value from each LOGICAL BLOCK GUARD field received in the single block of data from the data-out buffer into the corresponding LOGICAL BLOCK GUARD field of each logical block If the medium is formatted with type 3 protection information (see SBC-2): **a** If the ATO bit is set to one in the Control mode page (see 4.3.8), then the device server shall place each logical block application tag received in the single block of data into the corresponding LOGICAL BLOCK REFERENCE TAG field of each logical block. If the ATO bit is set to zero, then the device server may write any value into the LOGICAL BLOCK REFERENCE TAG field(s) of each logical block; and **b** The device server shall place the value from each LOGICAL BLOCK GUARD field received in the single block of data from the data-out buffer into the corresponding LOGICAL BLOCK GUARD field of each logical block.
0	1 [a]	The device server shall replace the first eight bytes of the block received from the data-out buffer to each physical sector with the physical address of the sector being written using the physical sector format (see 3.5.5.5).
1 [a]	0	The device server shall replace the first four bytes of the block received from the data-out buffer with the least significant four bytes of the LBA of the block being written, ending with the least significant byte (e.g., if the LBA is 77665544_33221100h, 33221100h is written with 33h written first and 00h written last).
1	1	The device server 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] If the medium is formatted with protection information then the protection information shall be written to a default value of FFFFFFFF_FFFFFFFFh in each of the written logical blocks.

NUMBER OF BLOCKS field

The NUMBER OF BLOCKS field specifies the number of contiguous logical blocks to be written, starting with the logical block specified by the LOGICAL BLOCK ADDRESS field. A NUMBER OF BLOCKS field set to zero specifies that the device server write all the logical blocks starting with the one specified in the LOGICAL BLOCK ADDRESS field to the last logical block on the medium. If the logical block address plus the number of blocks exceeds the capacity of the medium, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to LOGICAL BLOCK ADDRESS OUT OF RANGE.

3.74 WRITE SAME (16) command

The WRITE SAME (16) command (see table 210) requests that the device server transfer a single logical block from the data-out buffer and write the contents of that logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format. he WRITE SAME (16) command with the UNMAP bit supported is one of the possible commands that shall be implemented by device servers supporting thin provisioning (see SBC-3).E

Table 210. WRITE SAME (16) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (93h)							
1	WRPROTECT			Reserved	UNMAP	PBDATA	LBDATA	Reserved
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10	(MSB)	NUMBER OF BLOCKS						
13								(LSB)
14	Reserved			GROUP NUMBER				
15	CONTROL							

See the WRITE SAME (10) command (see 3.73) for the definitions of the fields in this command.

If the logical unit is thin provisioned (see SBC-3), then:

a) if the UNMAP bit is set to one, then the device server should perform an unmap operation (see SBC-3) on each LBA specified by the command instead of the specified write operation; or.
b) if the UNMAP bit is set to zero, then the device server shall perform the specified write operation.

The device server shall ignore the UNMAP bit 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:

a) the logical unit is fully provisioned (i.e., the TPE bit is set to zero in the READ CAPACITY (16) parameter data (see 3.26.2)); and
b) the UNMAP bit is set to one.

3.75 WRITE SAME (32) command

The WRITE SAME (32) command (see table 211) requests that the device server transfer a single logical block from the data-out buffer and write the contents of that logical block, with modifications based on the LBDATA bit and the PBDATA bit, to the specified range of logical block addresses. Each logical block includes user data and may include protection information, based on the WRPROTECT field and the medium format.The WRITE SAME (32) command with the UNMAP bit supported is one of the possible commands that shall be implemented by device servers supporting thin provisioning (see SBC-3).

Table 211. WRITE SAME (32) command

Bit / Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
5								
6	Reserved			GROUP NUMBER				
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)	SERVICE ACTION (000Dh)						
9								(LSB)
10	WRPROTECT			Reserved	UNMAP	PBDATA	LBDATA	Reserved
11	Reserved							
12	(MSB)	LOGICAL BLOCK ADDRESS						
19								(LSB)
20	(MSB)	EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG						
23								(LSB)
24	(MSB)	EXPECTED LOGICAL BLOCK APPLICATION TAG						
25								(LSB)
26	(MSB)	LOGICAL BLOCK APPLICATION TAG MASK						
27								(LSB)
28	(MSB)	NUMBER OF BLOCKS						
31								(LSB)

See the WRITE SAME (10) command (see 3.73) for the definitions of the GROUP NUMBER field, the WRPROTECT field, the PBDATA bit, the LBDATA bit, the LOGICAL BLOCK ADDRESS field, and the NUMBER OF BLOCKS field.

See the WRITE SAME (16) command (see 3.74) for the definitions of the UNMAP bit,

When checking of the LOGICAL BLOCK REFERENCE TAG field is enabled (see 3.73), the EXPECTED INITIAL LOGICAL BLOCK REFERENCE TAG field contains the value of the LOGICAL BLOCK REFERENCE TAG field expected in the protection information of the first logical block accessed by the command instead of a value based on the LBA (see SBC-3).

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is enabled (see 3.73), then the LOGICAL BLOCK APPLICATION TAG MASK field contains a value that is a bit mask for enabling the checking of the LOGICAL BLOCK APPLICATION TAG field in every instance of protection information for each logical block accessed by the command. A LOGICAL BLOCK APPLICATION TAG MASK bit set to one enables the checking of the corresponding bit of the EXPECTED LOGICAL BLOCK APPLICATION TAG field with the corresponding bit of the LOGICAL BLOCK APPLICATION TAG field in every instance of protection information.

If the ATO bit is set to one in the Control mode page (see 4.3.8) and checking of the LOGICAL BLOCK APPLICATION TAG field is disabled (see 3.73), or if the ATO bit is set to zero, then the LOGICAL BLOCK APPLICATION TAG MASK field and the EXPECTED LOGICAL BLOCK APPLICATION TAG field shall be ignored.

4.0 Parameters for all device types

This clause provides information on parameters for Diagnostic (Receive Diagnostic), Log (Sense), Mode (Sense), Protocol specific, and Vital product data (Inquiry) pages

4.1 Diagnostic parameters

4.1.1 Diagnostic page format and page codes for all device types

This subclause describes the diagnostic page structure and the diagnostic pages that are applicable to all SCSI devices. Diagnostic pages specific to each device type are described in the command standard that applies to that device type.

A SEND DIAGNOSTIC command with a PF bit set to one specifies that the SEND DIAGNOSTIC parameter list consists of a single diagnostic page and that the data returned by the subsequent RECEIVE DIAGNOSTIC RESULTS command that has the PCV bit set to zero shall use the diagnostic page format defined in table 212. A RECEIVE DIAGNOSTIC RESULTS command with a PCV bit set to one specifies that the device server return a diagnostic page using the format defined in table 212.

Table 212. Diagnostic page format

Bit Byte	7	6	5	4	3	2	1	0
0	PAGE CODE							
1	PAGE CODE SPECIFIC							
2	(MSB)	PAGE LENGTH (N-3)						
3								(LSB)
4	DIAGNOSTIC PARAMETERS							
n								

Each diagnostic page defines a function or operation that the device server shall perform as a result of a SEND DIAGNOSTIC command or the information being returned as a result of a RECEIVE DIAGNOSTIC RESULTS command with the PCV bit equal to one. The diagnostic parameters contain data that is formatted according to the page code specified.

PAGE CODE field

The PAGE CODE field identifies the diagnostic page (see table 213).

Table 213. Diagnostic page codes

Page Code	Diagnostic Page Name	Reference
00h	Supported Diagnostic Pages	4.1.2
01h - 2Fh	Defined by SES-2 for: **a** Enclosure services devices (i.e., SCSI devices with the PERIPHERAL DEVICE TYPE field set to 0Dh in standard INQUIRY data); and **b** SCSI devices with the ENCSERV bit set to one in standard INQUIRY data (see 3.6.2). **Note.** These pages are described in SES-2 these pages are passed along to any attached enclosure services device.	SES-2
30h - 3Eh	Reserved	
3Fh	See specific SCSI transport protocol for definition	
40h	Translate Address Input diagnostic page	
	Translate Address Output diagnostic page	
41h - 7Fh	See specific device type for definition	
80h - FFh	Vendor specific	

PAGE LENGTH field

The PAGE LENGTH field contains the length in bytes of the diagnostic parameters that follow this field. If the application client sends a SEND DIAGNOSTIC command with a parameter list containing a PAGE LENGTH field that results in the truncation of any parameter, then the 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 PARAMETER LIST.

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

4.1.2 Supported diagnostic pages (00h)

The Supported Diagnostic Pages diagnostic page (see table 214) returns the list of diagnostic pages implemented by the device server. This diagnostic page shall be implemented if the device server implements the diagnostic page format option of the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands.

Table 214. Supported diagnostic pages

Bit / Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (00h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (N-3)						
3								(LSB)
4	SUPPORTED PAGE LIST							
n								

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

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

PAGE LENGTH field

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

SUPPORTED PAGE LIST field

The SUPPORTED PAGE LIST field shall contain a list of all diagnostic page codes, one per byte, implemented by the device server in ascending order beginning with page code 00h.

4.1.3 Translate Address input page (40h)

Table 215 defines the Translate Address Input diagnostic page retrieved with the RECEIVE DIAGNOSTIC RESULTS command after the Translate Address Output diagnostic page (see 4.1.3) has been sent with the SEND DIAGNOSTIC command. If a Translate Address Output diagnostic page has not yet been processed, the results of a RECEIVE DIAGNOSTIC RESULTS command requesting this diagnostic page are vendor specific.

Table 215. Translate Address page—input diagnostic page (40h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (40h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	Reserved				SUPPLIED FORMAT			
5	RAREA	ALTSEC	ALTTK	Reserved	TRANSLATED FORMAT			
TRANSLATED ADDRESSes (if available)								
6	(MSB)	TRANSLATED ADDRESS 1						
:								
13								(LSB)
	…							
n-7	(MSB)	TRANSLATED ADDRESS x (if required)						
:								
n								(LSB)

PAGE CODE field

The PAGE CODE field is defined in 4.1.1 and shall be set to the value defined in table 215.

PAGE LENGTH field

The PAGE LENGTH field is defined in (see 4.1.1).

SUPPLIED FORMAT field

The SUPPLIED FORMAT field contains the value from the SEND DIAGNOSTIC command supplied format field (see Table 215).

RAREA (Reserved Area) bit

1 A Reserved Area (RAREA) bit of one indicates that all or part of the translated address falls within a reserved area of the medium (e.g., speed tolerance gap, alternate logical block, vendor reserved area, etc.). If the entire translated address falls within a reserved area the target may not return a translated address.

0 An RAREA bit of zero indicates that no part of the translated address falls within a reserved area of the medium.

TRANSLATED FORMAT field

The TRANSLATED FORMAT field contains the value from the TRANSLATE FORMAT field in the previous Translate Address Output diagnostic page (see 4.1.4).

Table 216. Address Field Logical Block Address Format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	LOGICAL BLOCK ADDRESS						
:								
3								(LSB)
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0

Table 217. Address Field Physical Sector Address Format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1	CYLINDER NUMBER							
2								(LSB)
3	HEAD NUMBER							
4	SECTOR NUMBER							
5								
6								
7								

ALTSEC (Alternate Sector) bit

1 An Alternate Sector (ALTSEC) bit of one indicates that the translated address is physically located in an alternate sector of the medium. If the drive cannot determine if all or part of the translated address is located in an alternate sector it shall set this bit to zero.

0 An ALTSEC bit of zero indicates that no part of the translated address is located in an alternate sector of the medium or that the drive is unable to determine this information.

ALTTRK (Alternate Track) bit

1 An Alternate Track (ALTTRK) bit of one indicates that part or all of the translated address is located on an alternate track of the medium or the drive cannot determine if all or part of the translated address is located on an alternate track.

0 An ALTTRK bit of zero indicates that no part of the translated address is located on an alternate track of the medium.

TRANSLATED ADDRESS field

The TRANSLATED ADDRESS field contains the address the target translated from the address supplied by the initiator in the SEND DIAGNOSTIC command. This field shall be in the format specified in the translate format field. The supported formats are shown in Tables and 217.

4.1.4 Translate Address Output diagnostic page (40h)

The Translate Address diagnostic pages allow the application client to translate an address in one of the formats supported by the FORMAT UNIT command (see 3.5.5.1) (i.e., a short block format address, a long block format address, a physical sector format address, or a bytes from index format address) into any one of the other formats. The address to be translated is sent to the device server with the SEND DIAGNOSTIC command and the results are returned to the application client by the RECEIVE DIAGNOSTIC RESULTS command.

Table 218 defines the format of the Translate Address Output diagnostic page sent with the SEND DIAGNOSTIC command. The translated address is returned in the Translate Address Input diagnostic page (see 4.1.3).

Table 218. Translate Address Output diagnostic page

Bit Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (40h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (000Ah)						
3								(LSB)
4	Reserved				SUPPLIED FORMAT			
5	Reserved				TRANSLATE FORMAT			
6	(MSB)	ADDRESS TO TRANSLATE						
.....								
13								(LSB)

PAGE CODE field

The PAGE CODE field is defined in 4.1.1 and shall be set to the value defined in table 218.

PAGE LENGTH field

The PAGE LENGTH field is defined in (see 4.1.1).

SUPPLIED FORMAT field

The SUPPLIED FORMAT field specifies the format of the ADDRESS TO TRANSLATE field. Valid values for this field are defined in the DEFECT LIST FORMAT of the FORMAT UNIT command (see 3.5). If the device server does not support the requested format it 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 PARAMETER LIST.

TRANSLATE FORMAT field

The TRANSLATE FORMAT field specifies the format the device server shall use for the result of the address translation. Valid values for this field are defined in the DEFECT LIST FORMAT field of the FORMAT UNIT command. If the device server does not support the specified format it shall terminate the SEND DIAGNOSTIC command with CHECK CONDI-

TION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

ADDRESS TO TRANSLATE field

The ADDRESS TO TRANSLATE field contains a single address descriptor which the application client is requesting the device server to translate. The format of this field depends on the value in the SUPPLIED FORMAT field. The formats are described in (see 3.5.5.1). If the short block format address descriptor is specified, the first four bytes of the ADDRESS TO TRANSLATE field shall contain the short block format address descriptor and the last four bytes shall contain 0000_0000h.

4.1.5 SCSI Enclosure Services pages (01h - 2Fh)

When the drive supports the SCSI Enclosure Service (SES) pages, it does not process the contents of the pages. The drive attempts to transfer the page contents to the enclosure using the Enclosure Services Interface (ESI) as directed by the command. Errors detected in the transfer are returned to the initiator in response to the command. See SCSI-3 Enclosure Services -2 (SES-2) for descriptions of the data presented in the SES pages. Reference the drive product manual to determine if the SES pages are supported.

Table 219. SCSI Enclosure Services page

Bit / Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (01h - 2Fh)							
1	Page Specific							
2	(MSB)	PAGE LENGTH (n - 3)						
3								(LSB)
4	Page Specific							
...								
n								

PAGE CODE

The PAGE CODE (01h - 2Fh) defined in SCSI Enclosure Services - 2 (SES-2) standard.

Page Specific

The Page Specific field is defined in the SCSI-3 Enclosure Services - 2 (SES-2) standard.

PAGE LENGTH

The PAGE LENGTH field is defined in (see 4.1.1).

4.2 Log parameters

4.2.1 Log page structure and page codes for all device types

This subclause describes the log page structure and the log pages that are applicable to all SCSI devices. Log pages specific to each device type are described in the command standard that applies to that device type. The LOG SELECT command (see 3.9) supports the ability to send zero or more log pages. The LOG SENSE command (see 3.10) returns a single log page specified in the PAGE CODE field of the CDB.

4.2.1.1 Log page format

Each log page begins with a four-byte page header followed by zero or more variable-length log parameters defined for that log page. The log page format is defined in table 220.

Table 220. Log page format

Bit / Byte	7	6	5	4	3	2	1	0
0	DS	SPF	PAGE CODE					
1	SUBPAGE CODE							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	Log parameter(s)							
4	Log parameter (First) (Length x)							
x+3								
	. .							
n-y+1	Log parameter (Last) (Length y)							
n								

DS (DISABLE SAVE) bit

- **0** For the LOG SENSE command (see 3.10), if the DS bit is set to zero, the log parameters are saved when the SP bit is set to one.
- **1** For the LOG SENSE command, if the DS bit is set to one, the log parameters are not saved. For the LOG SELECT command (see 3.9), the disable save (DS) bit operates in conjunction with the parameter code reset (PCR) bit, the save parameters (SP) bit, the page control (PC) field, and the PARAMETER LIST LENGTH field in the CDB.

SPF (SUBPAGE FORMAT) bit

- **0** If the SPF bit is set to zero, the SUBPAGE CODE field shall contain 00h.
- **1** If the SPF bit is set to one, the SUBPAGE CODE field shall contain a value between 01h and FFh.

PAGE CODE field

The value in the PAGE CODE field is the number of the log page being transferred.

SUBPAGE CODE field

The value in the SUBPAGE CODE field is the number of the subpage page being transferred.

If an application client specifies values in the PAGE CODE field and SUBPAGE CODE field for a log page that is reserved or not implemented by the logical unit, then the device server shall terminate the LOG SELECT 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 PARAMETER LIST LENGTH field in a LOG SELECT CDB contains zero, the meanings for the PCR bit, SP bit, and PC field are defined in table 60.

If the PARAMETER LIST LENGTH field in a LOG SELECT CDB contains a non-zero value (i.e., when parameter data is being sent with the LOG SELECT command), table 221 defines the meaning for the combinations of values for:

a) the PCR bit, the SP bit, and the PC field in the LOG SELECT CDB; and
a) the DS bit in the LOG SELECT parameter data.

Table 221. LOG SELECT PCR bit, SP bit, and DS bit meanings when parameter list length is not zero

PCR bit	SP bit	DS bit	Description
0b	0b	xb	The device server shall set the specified values[a] to the values in the parameter list and shall not save any values to non-volatile media.
0b	1b	0b	The device server shall set the specified values[a] to the values in the parameter list and shall process the optional saving of log parameter values as follows: a) If default data counter values are specified (see table 66), no values shall be saved; b) If values other than default data counter values are specified and the device server implements saving of the specified values[a], then the device server shall save the specified values[a] in the parameter list to non-volatile media; or c) If values other than default values are specified and the device server does not implements saving of one or more of the specified values[a], then the device server 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.
0b	1b	1b	The device server shall set the specified values[a] to the values in the parameter list and shall not save any values in the specified log page to non-volatile media.
1b	xb	xb	The device server 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] The specified parameters are determined by the format and linking field contents (see table 224) in the LOG SELECT parameter data and by the PC field contents (see table 66) in the LOG SELECT CDB.

PAGE LENGTH field

The value in the PAGE LENGTH field is the length in bytes of the following log parameters. If the application client sends a LOG SELECT command with a log page length that results in the truncation of any parameter, the device server 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.

4.2.1.2 Log parameter format

Most log pages contain one or more special data structures called log parameters (see table 222). Log parameters may be data counters of a particular event(s), the conditions under which certain operations were performed, or list parameters that contain a character string description of a particular event.

Table 222. Log parameter

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER CODE						
1								(LSB)
2	DU	Obsolete	TSD	ETC	TMC		FORMAT and LINKING	
3	PARAMETER LENGTH (N-3)							
4		PARAMETER VALUE						
n								

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

PARAMETER CODE field

The PARAMETER CODE field identifies the log parameter being transferred for that log page. If an application client specifies a value in the PARAMETER CODE field in the LOG SELECT command parameter data that is reserved or not implemented by the logical unit, then the device server 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.

DU bit, TSD bit, ETC bit, TMC field, and FORMAT AND LINKING

The DU bit, TSD bit, ETC bit, TMC field, and FORMAT AND LINKING field are collectively referred to as the parameter control byte. The bits and fields in the parameter control byte are described in 4.2.1.2.1.

PARAMETER LENGH

The PARAMETER LENGH field specifies the length in bytes of the parameter value field that follows. If the application client specifies a PARAMETER LENGH that results in the truncation of the parameter value field, the 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 PARAMETER LIST.

If an application client sends a log parameter that is not supported by the logical unit, the 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 PARAMETER LIST.

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

a) round to an acceptable value and terminate the command as described in 2.3; or
b) 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 parameter data for one LOG SELECT command contains more than one log page and the log pages are not in ascending order by page code value then subpage code value, then the device server 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 parameter data for one LOG SELECT command contains more than one log parameter in any one log page and the log parameters are not in ascending order by parameter code value, then the device server 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.

Note. Application clients should send LOG SENSE commands prior to sending LOG SELECT commands to determine supported log pages and page lengths.

4.2.1.2.1 Parameter control byte

DU bit, TSD bit, ETC bit, TMC field, and FORMAT AND LINKING field are collectively referred to as the parameter control byte. These fields are described in this subclause.

DU (Disable Update) bit

For cumulative log parameter values, indicated by the PC field (see table 66) of the LOG SELECT command and LOG SENSE command, the disable update (DU) bit is defined as follows:

0 DU set to zero indicates that the device server shall update the log parameter value to reflect all events that should be noted by that parameter; or

1 DU set to one indicates that the device server shall not update the log parameter value except in response to a LOG SELECT command that specifies a new value for the parameter.

Note. When updating cumulative log parameter values, a device server may use volatile memory to hold these values until a LOG SELECT or LOG SENSE command is received with an SP bit set to one or a vendor specific event occurs. As a result the updated cumulative log parameter values may be lost if a power cycle occurs.

The device server shall ignore the DU bit for threshold values, indicated by the PC field (see table 66) of the LOG SENSE command, or for list parameters as indicated by the FORMAT AND LINKING field received with a LOG SELECT command. The device server shall ignore the value of the DU bit in any such log parameters received with a LOG SELECT command.

TSD (Target Save Disable) bit

0 A target save disable (TSD) bit set to zero indicates that the logical unit implicitly saves the log parameter at vendor specific intervals. This implicit saving operation shall be done frequently enough to insure that the cumulative parameter values retain statistical significance (i.e., across power cycles).

1 A TSD bit set to one indicates that either the logical unit does not implicitly save the log parameter or implicit saving of the log parameter has been disabled individually by an application client setting the TSD bit to one. An application client may disable the implicit saving for all log parameters without changing any TSD bits using the GLTSD bit in the Control mode page.

ETC (Enable Threshold Comparison) bit

1 An enable threshold comparison (ETC) bit set to one indicates that a comparison to the threshold value is performed whenever the cumulative value is updated.

0 An ETC bit set to zero indicates that a comparison is not performed. The value of the ETC bit is the same for cumulative and threshold parameters.

TMC (Threshold Met Criteria) bit

The threshold met criteria (TMC) field (see table 223) defines the basis for comparison of the cumulative and threshold values. The TMC field is valid only if the ETC bit is set to one. The value of the TMC field is the same for cumulative and threshold parameters.

Table 223. Threshold met criteria

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

If the ETC bit is set to one and the result of the comparison is true, a unit attention condition shall be established for the initiator port associated with every I_T nexus, with the additional sense code set to THRESHOLD CONDITION MET.

FORMAT AND LINKING field

The FORMAT AND LINKING field (see table 224) indicates the type of log parameter and how parameters that reach their maximum value are handled.

TABLE 224. FORMAT AND LINKING field

Code	Log parameter type	Maximum value handling
00b	Data counter	If any other parameter in this log page reaches its maximum value, then this parameter shall stop incrementing until reinitialized by a LOG SELECT command.
01b	List format ASCII data	No maximum values to handle
10b	Data counter	If another parameter reported in this log page reaches its maximum value, then this parameter shall not stop incrementing. This parameter may be reinitialized by a LOG SELECT command.
11b	List format binary data	No maximum values to handle

A FORMAT AND LINKING field set to 00b or 10b indicates that the parameter is a data counter. Data counters are saturating counters associated with one or more events. A data counter is incremented whenever one of these events occurs. If a data counter has associated with it a vendor specific maximum value, then upon reaching this maximum value, the data counter shall not be incremented (i.e., its value does not wrap). When a data counter reaches its maximum value, the device server shall set the associated DU bit to one and handle other data counters in the log page as defined in table 224. If the data counter is at or reaches its maximum value during the processing of a command, the device server shall complete the command. If the command completes without error, except for the data counter being at its maximum value, and if the RLEC bit of the Control mode page (see 4.3.8) is set to one, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to RECOVERED ERROR, and the additional sense code set to LOG COUNTER AT MAXIMUM.

A FORMAT AND LINKING field set to 01b or 11b indicates that the parameter is a list parameter. If the FORMAT AND LINKING field set to 01b or 11b, the ETC bit and TMC field shall be set to zero. If the FORMAT AND LINKING field set to 01b or 11b and either the ETC bit and TMC field shall is set to a non-zero value in LOG SELECT parameter data, then the device server shall terminate the command with CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN PARAMETER LIST.

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

a) the parameter updated last shall have a higher parameter code than the parameter updated previously, except as defined in rule b); and
a) when the maximum parameter code value supported by the logical unit is reached, the device server shall assign the lowest parameter code value to the next log parameter (i.e., wrap-around parameter codes). If the associated LOG SELECT command completes without error, except for the parameter code being at its maximum value, and if the RLEC bit of the Control mode page (see 4.3.8) is set to one, then the command shall be terminated with CHECK CONDITION status, with the sense key set to RECOVERED ERROR, and the additional sense code set to LOG LIST CODES EXHAUSTED.

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

If a LOG SELECT command's parameter data contains a FORMAT AND LINKING value that is not allowed (see table 225) based on the FORMAT AND LINKING field value returned by a LOG SENSE command, the LOG SELECT 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 PARAMETER LIST.

Table 225. Allowed LOG SELECT FORMAT AND LINKING field values

FORMAT AND LINKING values returned by LOG SENSE	LOG SELECT format and linking values			
	00b	**01b**	**10b**	**11b**
00b	Allowed	Not Allowed	Not Allowed [a]	Not Allowed
01b	Not Allowed	Allowed	Not Allowed	Not Allowed
10b	Not Allowed [a]	Not Allowed	Allowed	Not Allowed
11b	Not Allowed	Not Allowed	Not Allowed	Allowed

[a] Seagate products do not allow changing FORMAT and LINKING.

The value for the FORMAT AND LINKING field for each log parameter are defined in the subclause that describes each log parameter.

If the value in the FORMAT AND LINKING field is 00b or 10b (i.e., the log parameter is a data counter parameter), then the values for the bits and fields in the parameter control byte for a LOG SENSE command or LOG SELECT command are described in 3.29.

If the value in the FORMAT AND LINKING field is 01b or 11b (i.e., the log parameter is a list parameter), then the values for the bits and fields in the parameter control byte for a LOG SENSE command or LOG SELECT command are described in 4.2.1.4.

4.2.1.3 Parameter control byte values for data counter parameters

If the FORMAT AND LINKING field is set to 00b or 10b (i.e., the parameter is a data counter parameter), the values for the bits and fields in the parameter control byte are shown in table 226.

Table 226. Parameter control byte values for data counter parameters

Field or bit	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0 or 1	0 or 1	When the DU bit is set to zero, the device server shall update the log parameter value to reflect all events that should be noted by that parameter. When the DU bit is set to one, the device server shall not update the log parameter value except in response to a LOG SELECT command that specifies a new value for the parameter.
TSD	0 or 1	0 or 1	When the TSD bit is set to zero, the device server shall save the log parameter to its medium at vendor specific intervals. When the TSD bit is set to one, implicit saving of the log parameter is disabled by an application client.
ETC	0 or 1	0 or 1	When the ETC bit is set to one, a comparison to the threshold value is performed whenever the cumulative value is updated. When the ETC bit is set to zero, a comparison is not performed.
TMC	any	any	The TMC field (see table 223) defines the basis for comparison of the cumulative and threshold values. The TMC field is valid only if the ETC bit is set to one.
FORMAT AND LINKING	00b or 10b	00b or 10b	The log parameter is a data counter (see table 224).

4.2.1.4 Parameter control byte values for list parameters

If the FORMAT AND LINKING field is set to 01b or 11b (i.e., the parameter is a list parameter), the values for the bits and fields in the parameter control byte are shown in table 227.

Table 227. Parameter control byte values for list parameters

Field or bit	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0 or 1	ignored	The DU bit is not defined for list parameters. The DU bit shall be set to zero when read with the LOG SENSE command, and shall be ignored when written with the LOG SELECT command.
TSD	0 or 1	0 or 1	When the TSD bit is set to zero, the device server shall save the log parameter to its medium at vendor specific intervals. When the TSD bit is set to one, implicit saving of the log parameter is disabled by an application client.
ETC	0 or 1	ignored	The ETC bit is not defined for list parameters. The ETC bit shall be set to zero when read with the LOG SENSE command, and shall be ignored when written with the LOG SELECT command.
TMC	any	ignored	The TMC field is not defined for list parameters. The TMC field shall be set to zero when read with the LOG SENSE command, and shall be ignored when written with the LOG SELECT command.
FORMAT AND LINKING	01b	01b	The log parameter is an ASCII format list parameter (see table 223).
	11b	11b	The log parameter is an binary format list parameter (see table 224).

The page code assignments for the log pages are listed in table 228.

Table 228. Log page codes

Page Code	Log Page Name	Reference
0Fh	Application Client	4.2.2
01h	Buffer Over-Run/Under-Run	4.2.6
2Fh	Informational Exceptions	4.2.8
0Bh	Last *n* Deferred Errors or Asynchronous Events	
07h	Last *n* Error Events	
06h	Non-Medium Error	4.2.9
18h	Protocol Specific Port	4.2.10
03h	Read Error Counter	4.2.6
04h	Read Reverse Error Counter	4.2.6
10h	Self-Test Results	4.2.10
0Eh	Start-Stop Cycle Counter	4.2.11
00h	Supported Log Pages	4.2.12
0Dh	Temperature	4.2.13
05h	Verify Error Counter	4.2.6
02h	Write Error Counter	4.2.6
08h - 0Ah	Reserved (may be used by specific device types)	
0Ch	Reserved (may be used by specific device types)	
11h - 14h	Reserved (may be used by specific device types)	
15h	Background Scan Results	4.3.2
16h - 17h	Reserved (may be used by specific device types)	
19h - 2Eh	Reserved (may be used by specific device types)	
3Fh	Reserved	
30h - 3Eh	Vendor specific	

4.2.2 Application Client log page (0Fh)

The Application Client log page (see table 229) provides a place for application clients to store information. The page code for the application client page is 0Fh.

Table 229. Application client log page

Bit Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (0Fh)					
1	SUBPAGE CODE (00h)							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	Application client log parameters							
4	First application client log parameter							
	. . .							
	Last application client log parameter							
n								

The PAGE CODE and PAGE LENGTH fields are described in 4.2.1.

Parameter codes 0000h through 0FFFh are for general usage application client data. The intended use for this information is to aid in describing the system configuration and system problems, but the specific definition of the data is application client specific. The general usage application client data parameters all have the format shown in table 230.

Table 230. General usage application client parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER CODE						
1								(LSB)
2	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
3	PARAMETER LENGTH (FCh)							
4	GENERAL USAGE PARAMETER BYTES							
255								

For general usage application client data, the value in the PARAMETER CODE field shall be between 0000h and 0FFFh. The first supported general usage application client parameter code shall be 0000h and additional supported parameters shall be sequentially numbered. If any general usage parameter codes are implemented, the device shall support at least 64 general usage parameter descriptors and they shall be parameter codes 0000h through 003Fh.

For the general usage application client parameter, the PARAMETER LENGTH value for each parameter shall be FCh.

The state of the log parameter control bits for parameters 0000h through 0FFFh is specified in table 231.

Table 231. Parameter control bits for general usage parameters (0000h through 0FFFh)

Bit	Value	Description
DU	1	Value provided by application client
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	11b	The parameter is a list parameter in binary format

The values stored in the GENERAL USAGE PARAMETER BYTES represent data sent to the device server in a previous LOG SELECT command. If a previous LOG SELECT command has not occurred, the data is vendor specific.

In the application client log page, parameter codes 1000h through FFFFh are reserved.

4.2.3 Background Scan Results log page (15h)

The Background Scan Results log page (see table 232) returns the background scanning status parameter and zero or more Medium Scan parameters when background scanning is supported. The Background Scanning Status parameter provides information about background pre-scan and background medium scan operations. Each Background Medium Scan Medium error log entry corresponds to a logical block where an error was detected. If the Background Scan Results log page is filled up, a new Background Medium Scan parameter overwrites the oldest entry. When a LOG SELECT command with PCR bit set to one is processed all Background Medium Scan parameters are deleted, however, the values in the Background Scanning Status parameter shall not be affected.

Table 232. Background Scan Results log page

Bit Byte	7	6	5	4	3	2	1	0
0	DS (1)	SPF (0)	PAGE CODE (15h)					
1	SUBPAGE CODE(00h)							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	Background Scan Results log parameters							
4		Background Scanning Status Parameters (see table 234)						
19								
	Background Scan parameter list							
20	(MSB)	Background Scan parameter (first) (see table y)						
43								(LSB)
. . .								
n-23	(MSB)	Background Scan parameter (last) (see table z)						
n								(LSB)

Table 233 defines the parameter codes for the Background Scan Results log page.

Table 233. Background Scan Results log page parameter codes

Parameter code	Description
0000h	Background scanning status parameter
0001h - 0800h	Background scan parameter
07FFhh - 7FFFh	Reserved

The background Scanning Status parameter (see table 234) contains status information about the background pre-scan and background medium scan features.

Table 234. Background scanning status parameter format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER CODE (0000h)						
1								(LSB)
2	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
3	PARAMETER LENGTH (0Ch)							
4	(MSB)	ACCUMULATED POWER ON MINUTES						
7								(LSB)
8	Reserved							
9	BACKGROUND SCANNING STATUS							
10	(MSB)	NUMBER OF BACKGROUND SCANS PERFORMED						
11								(LSB)
12	(MSB)	BACKGROUND MEDIUM SCAN PROGRESS						
13								(LSB)
14		NUMBER OF BACKGROUND SCANS PERFORMED						
15								

Table 235 defines the values for the log parameter control bits (see SPC-4) for this log parameter.

Table 235. Parameter control bits for Background Scanning Status log parameter

Field	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0b	any (i.e., ignored)	The DU bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
TSD	any	any	When the TSD bit is set to zero, the device server shall save the log parameter to its medium at vendor specific intervals. When the TSD bit is set to one, implicit saving of the log parameter is disabled by anapplication client.
ETC	0	any (i.e., ignored)	The ETC bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
TMC	00b	any (i.e., ignored)	The TMC field is not defined for list parameters, so shall be set to 00b when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
FORMAT AND LINKING	11b	11b	The log parameter is a binary list parameter.

PARAMETER LENGTH field

The PARAMETER LENGTH field indicates that number of bytes remaining in the log parameter.

ACCUMULATED POWER ON MINUTES field

The ACCUMULATED POWER ON MINUTES field indicates the number of minutes the device server has been powered on since manufacturing.

Table 236 defines the BACKGROUND SCANNING STATUS field.

Table 236. Background Scanning Status field

Code	Description
00h	No background scans active
01h	Background medium scan is active
02h	Background pre-scan is active
03h	Background medium scan halted due to fatal error
04h	Background medium scan halted due to a vendor-specific pattern of errors
05h	Background medium scan halted due to medium formatted without P-list
06h	Background medium scan halted - vendor-specific cause
07h	Background medium scan halted due to temperature out of allowed range
08h	Background medium scan halted, waiting for Background Medium Interval timer expiration.
09h - FFh	Reserved

NUMBER OF SCANS PERFORMED field

The NUMBER OF SCANS PERFORMED field indicates the number of background scans (i.e., the total number of background pre-scan operations plus the number of background medium scan operations) that have been performed since the SCSI target device was shipped by the manufacturer.

BACKGROUND MEDIUM SCAN PROGRESS field

The BACKGROUND MEDIUM SCAN PROGRESS field indicates the percent complete of a background scan operation in progress. The returned value is a numerator that has 65 536 (i.e., 1_0000h) as its denominator. If there is no background scan operation in progress (i.e., no background scan operation has been initiated since power on or the most recent background scan operation has completed), then the device server shall set the BACKGROUND MEDIUM SCAN PROGRESS field to 0000h.

NUMBER OF BACKGROUND MEDIUM SCAN PERFORMED

The NUMBER OF BACKGROUND MEDIUM SCAN PERFORMED field indicates the number of background medium scan operations that have been performed since the SCSI target device was shipped by the manufacturer. If the NUMBER OF BACKGROUND MEDIUM SCAN PERFORMED field contains 0000h, then the number of background medium scan operations is not reported.

A Background Medium Scan parameter (see table 237) describes a defect location on the medium that was encountered by background scanning.

Table 237. Background Medium Scan parameter format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER CODE (0001h - 0800h)						
1								(LSB)
2	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
3	PARAMETER LENGTH (14h)							
4	(MSB)	ACCUMULATED POWER ON MINUTES						
7								(LSB)
8	REASSIGN STATUS				SENSE KEY			
9	ADDITIONAL SENSE CODE							
10	ADDITIONAL SENSE CODE QUALIFIER							
11	VENDOR-SPECIFIC							
15								
16	(MSB)	LOGICAL BLOCK ADDRESS						
23								(LSB)

Table 238 defines the values for the log parameter control bits (see SPC-4) for this log parameter.

Table 238. Parameter control bits for Background Medium Scan log parameter

Field	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0b	any (i.e., ignored)	The DU bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
TSD	any	any	When the TSD bit is set to zero, the device server shall save the log parameter to its medium at vendor specific intervals. When the TSD bit is set to one, implicit saving of the log parameter is disabled by anapplica-tion client.
ETC	0	any (i.e., ignored)	The ETC bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
TMC	00b	any (i.e., ignored)	The TMC field is not defined for list parameters, so shall be set to 00b when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
FORMAT AND LINKING	11b	11b	The log parameter is a binary list parameter.

PARAMETER LENGTH field

The PARAMETER LENGTH field indicates the number of bytes remaining in the log parameter.

ACCUMULATED POWER ON MINUTES field

The ACCUMULATED POWER ON MINUTES field indicates the number of minutes the device server has been powered on since manufacturing at the time the background scan error occurred.

Table 239 defines the REASSIGN STATUS field.

Table 239. REASSIGN STATUS FIELD

Code	Error logging a	Description
0h	No	Reserved.
1h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and reassignment is pending receipt of:[bc] a) a command performing a write operation, if auto write reallocation is allowed (i.e., the AWRE bit is set to one in the Read-Write Error Recovery mode page (see 4.3.15); or b) a REASSIGN BLOCKS command.
2h	No	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was successfully reassigned by the device server with recovered data.
3h		Reserved
4h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and reassign by the device server failed and the logical block may or may not have an uncorrectable error.
5h	No	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was recovered by the device server via a rewrite in place.
6h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed, was successfully reassigned by the application client, and contains valid data (e.g., by a REASSIGN BLOCKS command that successfully recovered the data, or by a command performing a write operation). [c]
7h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed, was successfully reassigned by the application client, but does not contain valid data (e.g. by a REASSIGN BLOCKS command that did not successfully recover the data).[c]
8h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was not successfully reassigned by the application (e.g., by a REASSIGN BLOCKS command that failed).
9h - Fh		Reserved.

[a] If the entry in the column is "No", or the LOWIR bit is set to zero in the Background Control mode page (see 4.3.6), then the device server shall not create a Background Medium Scan parameter for the error. If the entry in the column is "Yes", and the LOWIR bit is set to one in the Background Control mode page, then the device server shall create a Background Medium Scan parameter for the error.

[b]The application client should use a command performing a write operation if it knows what data belongs on the logical block (e.g., in a redundancy group (see SBC-3), it uses data regenerated from the data on the other logical units in the redundancy group). The REASSIGN BLOCKS command may not be able to recover the data and does not report whether or not it successfully does so.

[c]The REASSIGN STATUS field in a given log parameter changes from 1h or 4h to 6h, 7h, or 8h when the logical block is reassigned, rewritten, or failed (i.e., the data in the reassigned block is not valid). If the logical block is reassigned or rewritten, any subsequently medium error to the logical block is reported in a new log parameter with the same value in the LOGICAL BLOCK ADDRESS field.

SENSE KEY, ADDITIONAL SENSE CODE, and the ADDITIONAL SENSE CODE QUALIFIER fields

The SENSE KEY field, ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field may contain a hierarchy of additional information relating to error conditions that occurred during background scanning. They are represented in the same format used by the sense data (see SPC-4).

LBA field

The LOGICAL BLOCK ADDRESS field indicates the LBA associated with the medium error.

4.2.4 Format Status log page

The Format Status log page (log page code 08h) captures the state of the direct-access block device since the most recent successful FORMAT UNIT command was completed. Additionally, this log page provides Defect Management information for the device server. The Format Status log page uses the log page format defined in SPC-4. Table 240 defines the parameter codes for the Format Status log page.

Table 240. Format Status log page parameter codes

Parameter code	Description
0000h	Format Data Out
0001h	Grown Defects During Certification
0002h	Total Blocks Reallocated During Format
0003h	Total New BLocks Reallocated
0004h	Power On Minutes Since Format
0005h - 7FFFh	Reserved
8000h - FFFFh	Vendor-specific

The PARAMETER LENGTH field of each log parameter (see SPC-4) contains the length of the corresponding PARAMETER VALUE field and is vendor-specific.

Event counts are returned as a result of the LOG SENSE command. The default value for each event count listed in table 240 shall be zero. Attempts to change these event counts by issuing a LOG SELECT with these fields set to non-zero values is not considered an error and shall have no effect on the saved values.

If information about a log parameter is not available, the device server shall return a value with each byte set to FFh (e.g., if the PARAMETER LENGTH field is set to 02h, the PARAMETER VALUE field is set to FFFFh). If the most recent FORMAT UNIT command failed, the device server shall return a value with each byte set to FFh for each log parameter. The Format Data Out parameter contains the entire FORMAT UNIT parameter list from the most recent successful FORMAT UNIT command. This includes:

a. the parameter list header;

b. the initialization pattern descriptor, if any; and

c. the defect list, if any.

The Grown Defects During Certification parameter is a count of the number of defects detected as a result of performing certification during processing of the most recent successful FORMAT UNIT command. This count reflects only those defects detected and replaced that were not already part of the PLIST or GLIST. If a certification pass was not performed the GROWN DEFECTS DURING CERTIFICATION field shall be set to zero.

The Total Blocks Reallocated During Format parameter is a count of the total number of logical blocks that were reallocated during the most recent successful FORMAT UNIT command.

Upon receiving the FORMAT UNIT command, the device server should set all parameters within the Format Status log page to indicate that no such information is available. Only upon successful completion of the FORMAT UNIT command should the device server update the affected fields.

The target save disable (TSD) bit in the PARAMETER CONTROL byte (see SPC-4) shall always be set to zero to indicate that the device server provides an implicit saving frequency.

Note. Removable media device servers may save log page information with the medium in a vendor-specific manner and location.

4.2.5 Cache Statistics page (37h)

Log Page code 37h specifies Cache Statistics page. The page format is shown in Table 241.

Table 241. Cache Statistics page (37h)

Bit Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (00h)					
1	SUBPAGE CODE(00h)							
2	PARAMETER CODE [1]							

[1] Parameter codes 00h through 04h are described following:

00h	This parameter code represents the number of logical blocks that have been sent to a SCSI initiator port.
01h	This parameter code represents the number of logical blocks that have been received from a SCSI initiator port.
02h	This parameter code represents the number of logical blocks read from the cache memory that have been sent to a SCSI initiator port.
03h	This parameter code represents the number of READ and WRITE commands that had data lengths equal or less than the current segment size.
04h	This parameter code represents the number of READ and WRITE commands that had data lengths greater that the current segment size.

4.2.6 Error counter log pages (WRITE, READ, and VERIFY, 02h, 03h, and 05h)

This subclause defines the error counter log pages (see table 242).

Table 242. Error counter log page codes

Page Code	Loge Page Name
03h	Read Error Counter
05h	Verify Error Counter
02h	Write Error Counter

The log page format is defined in 4.2.1. A log page may return one or more log parameters that record events defined by the parameter codes. Table 243 defines the parameter codes for the error counter log pages.

Table 243. Parameter codes for error counter log pages

Parameter code	Description
0000h	Errors corrected without substantial delay. An error correction was applied to get perfect data (a.k.a., ECC on-the-fly). “Without Substantial Delay” means the correction did not postpone reading of later sectors (e.g., a revolution was not lost). The counter is incremented once for each logical block that requires correction. Two different blocks corrected during the same command are counted as two events.
0001h	Errors corrected with possible delays. An error code or algorithm (e.g.,ECC, checksum) is applied in order to get perfect data with substantial delay. “With possible delay” means the correction took longer than a sector time so that reading/writing of subsequent sectors was delayed (e.g, a lost revolution). The counter is incremented once for each logical block that requires correction. A block with a double error that is correctable counts as one event and two different blocks corrected during the same command count as two events.
0002h	Total (e.g., rewrites or rereads). This parameter code specifies the counter counting the number of errors that are corrected by applying retries. This counts errors recovered, not the number of retries. If five retries were required to recover one block of data, the counter increments by one, not five. The counter is incremented once for each logical block that is recovered using retries. If an error is not recoverable while applying retries and is recovered by ECC, it isn’t counted by this counter; it will be counted by the counter specified by parameter code 01h–Error Corrected With Possible Delay.
0003h	Total errors corrected. This counter counts the total of parameter code errors 00h, 01h, and 02h. There is to be no “double counting” of data errors among these three counters. The sum of all correctable errors can be reached by adding parameter code 01h and 02h errors, not by using this total.
0004h	Total times correction algorithm processed. This parameter code specifies the counter that counts the total number of retries, or “times the retry algorithm is invoked.” If after five attempts a counter 02h type error is recovered, then five is added to this counter. If three retries are required to get a stable ECC syndrome before a counter 01h type error is corrected, then those three retries are also counted here. The number of retries applied to unsuccessfully recover an error (counter 06h type error) are also counted by this counter.
0005h	Total bytes processed. This parameter code specifies the counter that counts the total number of bytes either successfully or unsuccessfully read, written, or verified (depending on the log page) from the drive. If a transfer terminates early because of an unrecoverable error, only the logical blocks up to and including the one with the unrecoverable error are counted.
0006h	Total uncorrected errors. This parameter code specifies the counter that contains the total number of blocks for which an uncorrected data error has occurred.
0007h - 7FFFh	Reserved
8000h - FFFFh	Vendor specific

Note. The exact definition of the error counters is not part of this manual. These counters should not be used to compare products because the products may define errors differently.

4.2.7 Factory Log page (3Eh)

Log page code 3Eh specifies factory status parameters (see Table 244).

Table 244. Factory Log page (3Eh)

Bit / Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (3Eh)					
1	SUBPAGE CODE(00h)							
2	PARAMETER CODE [1] [2]							

[1] PARAMETER CODE 0000h–Power-on Time. This parameter code represents the number of drive power-on minutes. Currently the Power-on Time parameter (0000h) is the only parameter in this Log Page that is visible to OEM/customers.

[2] PARAMETER CODE 08h. This parameter reports the time, in minutes, to the next scheduled interrupt for a S.M.A.R.T. measurement.

4.2.8 Informational Exceptions log page (2Fh)

The Informational Exceptions log page (see table 245) provides a place for reporting detail about informational exceptions. The page code for the Informational Exceptions log page is 2Fh.

Table 245. Informational Exceptions log page

Bit / Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (2Fh)					
1	SUBPAGE CODE(00h)							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	INFORMATIONAL EXCEPTIONS LOG PARAMETERS							
	FIRST INFORMATIONAL EXCEPTIONS LOG PARAMETER							
	. . .							
n	LAST INFORMATIONAL EXCEPTIONS LOG PARAMETER							

PAGE CODE and PAGE LENGTH fields

The PAGE CODE and PAGE LENGTH fields are described in 4.2.1.

Table 246 defines the parameter codes.

Table 246. Informational exceptions parameter codes

Parameter code	Description
0000h	Informational exceptions general parameter data
0001h - FFFFh	Vendor specific

The informational exceptions general parameter data page has the format shown in table 247.

Table 247. Informational exceptions general parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER CODE (0000h)						
1								(LSB)
2	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
3	PARAMETER LENGTH (N-3)							
4	INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE							
5	INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER							
6	MOST RECENT TEMPERATURE READING							
7	VENDOR HDA TEMPERATURE TRIP POINT							
8	MAXIMUM TEMPERATURE							
9	Vendor specific							
11								

The values of the log parameter control bits for self test results log parameters are specified in table 248.

Table 248. Parameter control bits for Informational exceptions log parameter (0000h)

Bit	Value	Description
DU	0	Value provided by device server
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	11	The parameter list is in binary format

PARAMETER LENGTH field

The PARAMETER LENGTH field is described in 4.2.1. The parameter length shall be at least 04h.

INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE field

If the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE field contains zero, no informational exception condition is pending and contents of the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER field are unspecified. If the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE field contains any value other than zero, an informational exception condition exists that has an additional sense code indicated by INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE field and an ADDITIONAL SENSE CODE QUALIFIER indicated by the INFORMATIONAL EXCEPTION ADDITIONAL SENSE CODE QUALIFIER field.

MOST RECENT TEMPERATURE READING field

The MOST RECENT TEMPERATURE READING field indicates the temperature in degrees Celsius of the SCSI target device at the time the LOG SENSE command is performed. Temperatures equal to or less than zero degrees Celsius shall be indicated by a value of zero. If the device server is unable to detect a valid temperature because of a sensor failure or other condition, the value returned shall be FFh. The temperature should be reported with an accuracy of plus or minus three Celsius degrees while the device is operating at a steady state within the environmental limits specified for the device.

VENDOR HDA TEMPERATURE TRIP POINT field

This field indicates the drive HDA temperature, in degrees Celsius, at which the drive will post a status with Sense Key of RECOVERED ERROR or UNIT ATTENTION and ASC/ASCQ Warning—Specified Temperature Exceeded

MAXIMUM TEMPERATURE

Maximum Temperature field is the maximum temperature in degrees Celsius measured and logged by the device. This parameter value never decreases. The parameter content of this field ranges from 00h to FFh.

4.2.9 Non-Medium Error log page (06h)

The Non-Medium Error log page (page code 06h) provides for summing the occurrences of recoverable error events other than write, read, or verify failures. No discrimination among the various types of events is provided by parameter code (see table 249). Vendor specific discrimination may be provided through Seagate specific parameter codes.

Table 249. Non-medium error event parameter codes

Parameter code	Description
0000h	Non-medium error count
0001h - 7FFFh	Reserved
8000h - FFFFh	Vendor specific error counts

The FORMAT AND LINKING field of each log parameter shall be set to 00b or 10b indicating it contains a data counter.

4.2.10 Self-Test Results log page (10h)

The Self-Test Results log page (see table 250) provides the results from the 20 most recent self-tests (see SPC-4 clause 5.5). Results from the most recent self-test or the self-test currently in progress shall be reported in the first self-test log parameter; results from the second most recent self-test shall be reported in the second self-test log parameter; etc. If fewer than 20 self-tests have occurred, the unused self-test log parameter entries shall be zero filled.

Table 250. Self-Test Results log page

Bit Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (10h)					
1	SUBPAGE CODE(00h)							
2	(MSB)	PAGE LENGTH (190h)						
4								(LSB)
	SELF-TEST RESULTS LOG PARAMETERS							
 23	FIRST SELF-TEST RESULTS LOG PARAMETER (most recent)							
	. . .							
384 403	TWENTIETH SELF-TEST RESULTS LOG PARAMETER (least recent)							

DS bit, SPF bit, PAGE CODE field and PAGE LENGTH field

The DS bit, SPF bit, PAGE CODE field and PAGE LENGTH field are described in 4.2.1.

Table 251 shows the format of one self-test log parameter.

Table 251. Self-test results log parameter format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	PARAMETER CODE (0001h TO 0014h)						
1								(LSB)
2	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
3	PARAMETER LENGTH (10h)							
4	SELF-TEST CODE			Reserved	SELF-TEST RESULTS			
5	SELF-TEST NUMBER							
6	(MSB)	ACCUMULATED POWER ON HOURS						
7								(LSB)
8	(MSB)	ADDRESS OF FIRST FAILURE						
15								(LSB)
16	Reserved				SENSE KEY			
17	ADDITIONAL SENSE CODE							
18	ADDITIONAL SENSE CODE QUALIFIER							
19	Vendor specific							

PARAMETER CODE field

The PARAMETER CODE field identifies the log parameter being transferred. The PARAMETER CODE field for the results of the most recent self-test shall contain 0001h; the PARAMETER CODE field for the results of the second most recent test shall contain 0002h; etc.

The FORMAT AND LINKING field for each log parameter in the Self-Test Results log page shall be set to 11b, indicating that the parameters are binary format list parameters.The values of the log parameter control bits for self test results log parameters is specified in table 252.

Table 252. Parameter control bits for self-test results log parameters

Bit	Value	Description
DU	0	Value provided by device server
DS	0	Device server supports saving of parameter
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	11b	The parameter is in binary format

PARAMETER LENGTH field

The PARAMETER LENGTH field shall contain 10h.

SELF-TEST CODE field

The SELF-TEST CODE field contains the value in the SELF-TEST CODE field of the SEND DIAGNOSTIC command that initiated this self-test.

Table 253 defines the content of the SELF-TEST RESULTS field.

Table 253. SELF-TEST RESULTS field

Code	Description
0h	The self-test completed without error.
1h	The background self-test was aborted by the application client using a SEND DIAGNOSTIC command (see 3.49) with the SELF-TEST CODE field set to 100b (i.e., abort background self-test).
2h	The self-test routine was aborted by an application client using a method other than a SEND DIAGNOSTIC command with the SELF-TEST CODE field set to 100b (e.g., by a task management function, or by issuing an exception command as defined in SPC-4).
3h	An unknown error occurred while the device server was processing the self-test and the device server was unable to complete the self-test.
4h	The self-test completed with a failure in a test segment, and the test segment that failed is not known.
5h	The first segment of the self-test failed.
6h	The second segment of the self-test failed.
7h	Another segment of the self-test failed and which test is indicated by the contents of the SELF-TEST NUMBER field.
8h-Eh	Reserved
Fh	The self-test is in progress.

SELF-TEST NUMBER field

The SELF-TEST NUMBER field identifies the self-test that failed and consists of either:

a) The number of the segment that failed during the self-test; or

b) The number of the test that failed and the number of the segment in which the test was run, using a vendor specific method for placing the two values in the one field.

When the segment in which the failure occurred is not able to be identified or need not be identified, the SELF-TEST NUMBER field shall contain 00h.

ACCUMULATED POWER ON HOURS field

The ACCUMULATED POWER ON HOURS field contains the total hours for the device server has been powered on since manufacturing at the time the self-test is completed. If the test is still in progress, the content of the ACCUMULATED POWER ON HOURS field shall be zero. If the power-on hours for the device server at the time the self-test was completed is greater than FFFFh then the content of the ACCUMULATED POWER ON HOURS field shall be FFFFh.

ADDRESS OF FIRST FAILURE field

The ADDRESS OF FIRST FAILURE field contains information that locates the failure on the media. If the logical unit implements logical blocks, the content of the ADDRESS OF FIRST FAILURE field is the first logical block address where a self-test error occurred. This implies nothing about the quality of any other logical block on the logical unit, since the testing during which the error occurred may not have been performed in a sequential manner. This value shall not change (e.g., as the result of block reassignment). The content of the ADDRESS OF FIRST FAILURE field shall be FFFF FFFF FFFF FFFFh if no errors occurred during the self-test or if the error that occurred is not related to an identifiable media address.

SENSE KEY, ADDITIONAL SENSE CODE, and ADDITIONAL SENSE CODE QUALIFIER fields

The SENSE KEY field, ADDITIONAL SENSE CODE field, and ADDITIONAL SENSE CODE QUALIFIER field may contain a hierarchy of additional information relating to error or exception conditions that occurred during the self-test represented in the same format used by the sense data (see REQUEST SENSE command).

4.2.11 Start-Stop Cycle Counter log page (0Eh)

This subclause defines the Start-Stop Cycle Counter log page (page code 0Eh). A device server that implements the Start-Stop Cycle Counter log page shall implement one or more of the defined parameters. Table 254 shows the Start-Stop Cycle Counter log page with all parameters present.

Table 254. Start-Stop Cycle Counter log page (Sheet 1 of 2)

Bit Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0b)	PAGE CODE (0Eh)					
1	SUBPAGE CODE(00h)							
2	(MSB)	PAGE LENGTH (24h)						
3								(LSB)
4	(MSB)	PARAMETER CODE 0001H DATE OF MANUFACTURE						
5								(LSB)
6	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
7	PARAMETER LENGTH (06h)							

Table 254. Start-Stop Cycle Counter log page (Sheet 2 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)	YEAR OF MANUFACTURE (4 ASCII CHARACTERS)						
11								(LSB)
12	(MSB)	WEEK OF MANUFACTURE (2 ASCII CHARACTERS)						
13								(LSB)
14	(MSB)	PARAMETER CODE 0002H ACCOUNTING DATE						
15								(LSB)
16	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
17	PARAMETER LENGTH (06h)							
18	(MSB)	ACCOUNTING DATE YEAR (4 ASCII CHARACTERS)						
21								(LSB)
22	(MSB)	ACCOUNTING DATE WEEK (2 ASCII CHARACTERS)						
23								(LSB)
24	(MSB)	PARAMETER CODE 0003H SPECIFIED CYCLE COUNT OVER DEVICE LIFETIME						
25								(LSB)
26	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
27	PARAMETER LENGTH (04h)							
28	(MSB)	SPECIFIED CYCLE COUNT OVER DEVICE LIFETIME (4-BYTE BINARY NUMBER)						
31								(LSB)
32	(MSB)	PARAMETER CODE 0004H ACCUMULATED START-STOP CYCLES						
33								(LSB)
34	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
35	PARAMETER LENGTH (04h)							
36	(MSB)	ACCUMULATED START-STOP CYCLES (4-BYTE BINARY NUMBER)						
39								(LSB)

The year and week in the year that the SCSI target device was manufactured shall be contained in the parameter value of the log parameter in which the parameter code is 0001h. The date of manufacture shall not be saveable by the application client using the LOG SELECT command (i.e., the log parameter DS bit shall be set to one). The date is expressed in numeric ASCII characters (30h – 39h) in the form YYYYWW, as shown in table 254. For the log parameter in which the parameter code value is 0001h, the values of the parameter control bits are defined in table 255.

Table 255. Parameter control bits for date of manufacture parameter (0001h)

Bit	Value	Description
DU	0	Value provided by device server
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	01b	The parameter is a list parameter in ASCII format

The accounting date specified by parameter code 0002h may be saved using a LOG SELECT command to indicate when the device was placed in service. If the parameter is not yet set or is not settable, the default value placed in the parameter field shall be 6 ASCII space characters (20h). The field shall not be checked for validity by the device server.

For the log parameter in which the parameter code value is 0002h, the values of the parameter control bits are defined in table 256.

Table 256. Parameter control bits for accounting date parameter (0002h)

Bit	Value	Description
DU	0	Value provided by device server
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	01b	The parameter is a list parameter in ASCII format

The parameter value in the specified cycle count over device lifetime log parameter (parameter code 0003h) shall contain a four-byte binary value that indicates how many stop-start cycles may typically be performed over the lifetime of the SCSI target device without degrading the SCSI target device's operation or reliability outside the limits specified by the manufacturer of the SCSI target device. The specified cycle count over device lifetime parameter shall not be saveable by the application client using the LOG SELECT command (i.e., the log parameter DS bit shall be set to one). For the log parameter in which the parameter code value is 0003h, the values of the parameter control bits are defined in table 257.

Table 257. Parameter control bits for start-stop cycle counter parameters (0003h and 0004h)

Bit	Value	Description
DU	0	Value provided by device server
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	01b	The parameter is a list parameter in binary format

The parameter value in the accumulated start-stop cycles log parameter (parameter code 0004h) shall contain a four-byte binary value that indicates how many stop-start cycles the SCSI target device has detected since its date of manufacture. The accumulated start-stop cycles parameter shall not be saveable by the application client using the LOG SELECT command (i.e., the log parameter DS bit shall be set to one). The time at which the count is incremented during a start-stop cycle is vendor specific. For rotating magnetic storage devices, a single start-stop cycle is defined as an operational cycle that begins with the disk spindle at rest, continues while the disk accelerates to its normal operational rotational rate, continues during the entire period the disk is rotating, continues as the disk decelerates toward a resting state, and ends when the disk is no longer rotating. For devices without a spindle or with multiple spindles, the definition of a single start-stop cycle is vendor specific. The count is incremented by one for each complete start-stop cycle. No comparison with the value of parameter 0003h shall be performed by the device server. For the log parameter in which the parameter code value is 0004h, the values of the parameter control bits are defined in table 257.

4.2.12 Supported Log Pages log page (00h)

The Supported Log Pages log page (see table 258) returns the list of log pages implemented by the logical unit. Logical units that implement the LOG SENSE command shall implement this log page.

Table 258. Supported log pages

Bit Byte	7	6	5	4	3	2	1	0
0	DS	SPF(1b)	PAGE CODE (00h)					
1	SUBPAGE CODE (FFh)							
2	(MSB)	PAGE LENGTH (N-3)						
3								(LSB)
4	SUPPORTED PAGE / SUBPAGE LIST							
n								

This log page is not defined for the LOG SELECT command. This log page returns the list of supported log pages for the specified logical unit.

PAGE LENGTH field

The PAGE LENGTH field indicates the length in bytes of the following supported log page list.

SUPPORTED PAGE / SUBPAGE LIST field

The SUPPORTED PAGE / SUBPAGE LIST field shall contain a list of all log page codes implemented by the logical unit in ascending order beginning with page code 00h.

4.2.13 Temperature log page (0Dh)

This subclause defines the Temperature log page (page code 0Dh). A device server that implements the Temperature log page shall implement parameter 0000h and may implement parameter 0001h. Table 259 shows the Temperature log page with all parameters present.

Table 259. Temperature log page

Bit / Byte	7	6	5	4	3	2	1	0
0	DS	SPF (0)	PAGE CODE (0Dh)					
1	SUBPAGE CODE (00h)							
2	(MSB)	PAGE LENGTH (0Ch)						
3								(LSB)
4	(MSB)	PARAMETER CODE 0000H TEMPERATURE						
5								(LSB)
6	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
7	PARAMETER LENGTH (02h)							
8	Reserved							
9	TEMPERATURE (DEGREES CELSIUS)							
10	(MSB)	PARAMETER CODE 0001H REFERENCE TEMPERATURE						
11								(LSB)
12	DU	Obsolete	TSD	ETC	TMC		FORMAT AND LINKING	
13	PARAMETER LENGTH (02h)							
14	Reserved							
15	REFERENCE TEMPERATURE (DEGREES CELSIUS)							

The parameter value in the temperature log parameter (parameter code 0000h) shall contain a one-byte binary value that indicates the temperature of the SCSI target device in degrees Celsius at the time the LOG SENSE command is performed. Temperatures equal to or less than zero degrees Celsius shall be indicated by a value of zero. If the device server is unable to detect a valid temperature because of a sensor failure or other condition, then the value returned shall be FFh. The temperature should be reported with an accuracy of plus or minus three Celsius degrees while the SCSI target device is operating at a steady state within its environmental limits. No comparison is performed between the temperature value specified in parameter 0000h and the reference temperature specified in parameter 0001h. The state of the parameter control bits for parameter 0000h is specified in table 260.

Table 260. Parameter control bits for temperature parameters (0000h and 0001h)

Bit	Value	Description
DU	0	Value provided by device server
TSD	0	Device server manages saving of parameter
ETC	0	No threshold comparison is made on this value
TMC	xx	Ignored when the ETC bit is set to zero
FORMAT AND LINKING	11b	The parameter is a list parameter in binary format

A reference temperature for the device may be returned by the device server as follows:

(a) If a reference temperature is returned, the parameter value in the reference temperature log parameter (parameter code 0001h) shall contain a one-byte binary value that indicates the maximum reported sensor temperature in degrees Celsius at which the SCSI target device is capable of operating continuously without degrading the SCSI target device's operation or reliability beyond manufacturer accepted limits; or

(b) If no reference temperature is returned, then:

   a) The log parameter with parameter code 0001h may not be included in the log page; or

   b) The parameter value in the reference temperature log parameter (parameter code 0001h) may be set to FFh.

The reference temperature may change for vendor specific reasons. The state of the parameter control bits for parameter 0001h is specified in table 260.

4.3 Mode parameters

4.3.1 Mode parameters overview

This subclause describes the mode parameter headers, block descriptors, and mode pages used with MODE SELECT command (see 3.12 and 3.13) and MODE SENSE command (see 3.14 and 3.15) that are applicable to all SCSI devices. Subpages are identical to mode pages except that they include a SUBPAGE CODE field that further differentiates the mode page contents. Mode pages specific to each device type are described in the command standard that applies to that device type.

Note. Many of the mode parameters in the following pages are changeable. A MODE SENSE command with the PC bit set to one will return a mask indicating the mode parameters that may be changed by a SCSI initiator port. Seagate disc drive product manuals indicate which pages a drive supports, what the default mode values are, and what mode parameters may be changed.

4.3.2 Mode parameter list format

The mode parameter list shown in table 261 contains a header, followed by zero or more block descriptors, followed by zero or more variable-length mode pages. Parameter lists are defined for each device type.

Table 261. Mode parameter list

Bit Byte	7	6	5	4	3	2	1	0
	MODE PARAMETER HEADER							
	BLOCK DESCRIPTOR(S)							
	MODE PAGE(S) OR VENDOR SPECIFIC (E.G., PAGE CODE SET TO ZERO)							

4.3.3 Mode parameter header formats

The mode parameter header that is used by the MODE SELECT(6) command (see 3.11) and the MODE SENSE(6) command (see 3.13) is defined in table 262.

Table 262. Mode parameter header(6)

Bit Byte	7	6	5	4	3	2	1	0
0	MODE DATA LENGTH							
1	MEDIUM TYPE							
2	WP	Reserved		DPOFUA	Reserved			
3	BLOCK DESCRIPTOR LENGTH							

The mode parameter header that is used by the MODE SELECT(10) command (see 3.12) and the MODE SENSE(10) command (see 3.14) is defined in table 263.

Table 263. Mode parameter header(10)

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	MODE DATA LENGTH						
1								(LSB)
2	MEDIUM TYPE							
3	WP	Reserved		DPOFUA	Reserved			
4	Reserved							LONGLBA
5	Reserved							
6	(MSB)	BLOCK DESCRIPTOR LENGTH						
7								(LSB)

MODE DATA LENGTH field

When using the MODE SENSE command, the MODE DATA LENGTH field indicates the length in bytes of the following data that is available to be transferred. The mode data length does not include the number of bytes in the MODE DATA LENGTH field. When using the MODE SELECT command, this field is reserved.

Note. Logical units that support more than 256 bytes of block descriptors and mode pages may need to implement ten-byte mode commands. The mode data length field in the six-byte CDB header limits the returned data to 256 bytes.

MEDIUM TYPE field

The contents of the MEDIUM TYPE field are unique for each device type. Refer to the mode parameters subclause of the specific device type command standard for definition of these values. Some device types reserve this field.

WP (Write Protect) bit

1 A WP bit set to one indicates that the medium is write-protected. The medium may be write protected when the software write protect (SWP) bit in the Control mode page (see 4.3.8) is set to one or if another vendor specific mechanism causes the medium to be write protected.

0 A WP bit set to zero indicates that the medium is not write-protected.

DPOFUA (DPO and FUA support) bit

0 A DPOFUA bit set to zero indicates that the device server does not support the DPO and FUA bits.

1 When used with the MODE SENSE command, a DPOFUA bit set to one indicates that the device server supports the DPO and FUA bits (see 3.20)

LONGLBA (Long LBA) bit

0 If the Long LBA (LONGLBA) bit is set to zero, the mode parameter block descriptor(s), if any, are each eight bytes long.

1 If the LONGLBA bit is set to one, the mode parameter block descriptor(s), if any, are each sixteen bytes long.

BLOCK DESCRIPTOR LENGTH field

The BLOCK DESCRIPTOR LENGTH field contains the length in bytes of all the block descriptors. It is equal to the number of block descriptors times eight if the LONGLBA bit is set to zero or times sixteen if the LONGLBA bit is set to one, and does not include mode pages or vendor specific parameters (e.g., page code set to zero), if any, that may follow the last block descriptor. A block descriptor length of zero indicates that no block descriptors are included in the mode parameter list. This condition shall not be considered an error.

4.3.4 Mode parameter block descriptors

4.3.4.1 Mode block descriptors overview

If the device server returns a mode parameter block descriptor, it shall return a short LBA mode parameter block descriptor (see 4.3.4.2) in the mode parameter data in response to:

a) a MODE SENSE (6) command; or

b) a MODE SENSE (10) command with the LLBAA bit set to zero.

If the device server returns a mode parameter block descriptor and the number of logical blocks is greater than FFFFFFFFh, it may return a long LBA mode parameter block descriptor (see 4.3.4.3) in the mode parameter data in response to a MODE SENSE (10) command with the LLBAA bit set to one.

If the application client sends a mode parameter block descriptor in the mode parameter list, it shall send a short LBA mode parameter block descriptor (see 4.3.4.2) for a MODE SELECT (6) command.

If the application client sends a mode parameter block descriptor in the mode parameter list, it may send a long LBA mode parameter block descriptor (see 4.3.4.3) for a MODE SELECT (10) command.

Support for the mode parameter block descriptors is optional. The device server shall establish a unit attention condition with the additional sense code of MODE PARAMETERS CHANGED (see SPC-4 and SAM-4) when the block descriptor values are changed.

4.3.4.2 Short LBA mode parameter block descriptor

Table x defines the block descriptor for direct-access block devices used:

a. with the MODE SELECT (6) and MODE SENSE (6) commands, and
b. with the MODE SELECT (10) and MODE SENSE (10) commands when the LONGLBA bit is set to zero in the mode parameter header (see 4.3.3))

Table 264. Short LBA mode parameter block descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3								(LSB)
4	Reserved							
5	(MSB)							
	LOGICAL BLOCK LENGTH							
7								(LSB)

A device server shall respond to a MODE SENSE command (see SPC-4) by reporting the number of logical blocks specified in the NUMBER OF LOGICAL BLOCKS field sent in the last MODE SELECT command that contained a mode parameter block descriptor. If no MODE SELECT command with a mode parameter block descriptor has been received then the current number of logical blocks shall be returned. To determine the number of logical blocks at which the logical unit is currently formatted, the application client shall use the READ CAPACITY command (see 3.24) rather than the MODE SENSE command.

On a MODE SENSE command, the device server may return a value of zero indicating that it does not report the number of logical blocks in the short LBA mode parameter block descriptor.

On a MODE SENSE command, if the number of logical blocks on the medium exceeds the maximum value that is able to be specified in the NUMBER OF LOGICAL BLOCKS field, the device server shall return a value of FFFFFFFh.

If the logical unit does not support changing its capacity by changing the NUMBER OF LOGICAL BLOCKS field using the MODE SELECT command (see 3.11 and 3.12), the value in the NUMBER OF LOGICAL BLOCKS field is ignored.

If the device supports changing its capacity by changing the NUMBER OF LOGICAL BLOCKS field, then the NUMBER OF LOGICAL BLOCKS field is interpreted as follows:

a. If the NUMBER OF LOGICAL BLOCKS field is set to zero, the logical unit shall retain its current capacity if the logical block length has not changed. If the NUMBER OF LOGICAL BLOCKS field is set to zero and the content of the LOGICAL BLOCK LENGTH field (i.e., new logical block length) is different than the current logical block length, the logical unit shall be set to its maximum capacity when the new logical block length takes effect (i.e., after a successful FORMAT UNIT command);
b. If the NUMBER OF LOGICAL BLOCKS field is greater than zero and less than or equal to its maximum capacity, the logical unit shall be set to that number of logical blocks. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length, the logical unit shall not become format corrupt. This capacity setting shall be retained through power cycles, hard resets, logical unit resets, and I_T nexus losses. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length this capacity setting shall take effect on successful completion of the MODE SELECT command. If the content of the LOGICAL BLOCK LENGTH field (i.e., new logical block length) is different than the current logical block length this capacity setting shall take effect when the new logical block length takes effect (i.e., after a successful FORMAT UNIT command);
c. If the NUMBER OF LOGICAL BLOCKS field is set to a value greater than the maximum capacity of the device and less than FFFFFFFFh, then the MODE SELECT 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 PARAMETER LIST. The logical unit shall retain its previous logical block descriptor settings; or
d. If the NUMBER OF LOGICAL BLOCKS field is set to FFFFFFFFh, the logical unit shall be set to its maximum capacity. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length, the logical unit shall not become format corrupt. This capacity setting shall be retained through power cycles, hard resets, logical unit resets, and I_T nexus losses. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length this capacity setting shall take effect of successful completion of the MODE SELECT command. If the content of the LOGICAL BLOCK LENGTH field (i.e., new logical block length) is different than the current logical block length this capacity setting shall take effect when the new logical block length takes effect (i.e., after a successful FORMAT UNIT command).

The LOGICAL BLOCK LENGTH field specifies the length in bytes of each logical block. No change shall be made to any logical blocks on the medium until a format operation is initiated by an application client.

A device server shall respond to a MODE SENSE command (see 3.13 and 3.14) by reporting the length of the logical blocks as specified in the LOGICAL BLOCK LENGTH field sent in the last MODE SELECT command that contained a mode parameter block descriptor. If no MODE SELECT command with a block descriptor has been received then the current logical block length shall be returned (e.g., if the logical block length is 512 bytes and a MODE SELECT command occurs with the LOGICAL BLOCK LENGTH field set to 520 bytes, any MODE SENSE commands would return 520 in the LOGICAL BLOCK LENGTH field). To determine the logical block length at which the logical unit is currently formatted, the application client shall use the READ CAPACITY command rather than the MODE SELECT command.

4.3.4.3 Long LBA mode parameter block descriptor

Table 265 defines the block descriptor for direct-access block devices used with the MODE SELECT (10) command and MODE SENSE (10) command when the LONGLBA bit is set to one in the mode parameter header (see 4.3.3).

Table 265. Long LBA mode parameter block descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)	NUMBER OF LOGICAL BLOCKS						
7								(LSB)
8	Reserved							
11								
12	(MSB)	LOGICAL BLOCK LENGTH						
15								(LSB)

A device server shall respond to a MODE SENSE command (see 3.13 and 3.14) by reporting the number of logical blocks specified in the NUMBER OF LOGICAL BLOCKS field sent in the last MODE SELECT command that contained a mode parameter block descriptor. If no MODE SELECT command with a mode parameter block descriptor has been received then the current number of logical blocks shall be returned. To determine the number of logical blocks at which the logical unit is currently formatted, the application client shall use the READ CAPACITY command rather than the MODE SENSE command.

On a MODE SENSE command, the device server may return a value of zero indicating that it does not report the number of logical blocks in the long LBA mode parameter block descriptor.

If the logical unit does not support changing its capacity by changing the NUMBER OF LOGICAL BLOCKS field using the MODE SELECT command (see SPC-4), the value in the NUMBER OF LOGICAL BLOCKS field is ignored. If the device supports changing its capacity by changing the NUMBER OF LOGICAL BLOCKS field, then the NUMBER OF LOGICAL BLOCKS field is interpreted as follows:

a. If the NUMBER OF LOGICAL BLOCKS field is set to zero, the logical unit shall retain its current capacity if the logical block length has not changed. If the NUMBER OF LOGICAL BLOCKS field is set to zero and the content of the LOGICAL BLOCK LENGTH field (i.e., new logical block length) is different than the current logical block length, the logical unit shall be set to its maximum capacity when the new logical block length takes effect (i.e., after a successful FORMAT UNIT command).
b. If the NUMBER OF LOGICAL BLOCKS field is greater than zero and less than or equal to its maximum capacity, the logical unit shall be set to that number of logical blocks. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length, the logical unit shall not become format corrupt. This capacity setting shall be retained through power cycles, hard resets, logical unit resets, and I_T nexus losses. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length this capacity setting shall take effect on successful completion of the MODE SELECT command. If the content of the LOGICAL BLOCK LENGTH field (i.e., new logical block length) is different than the current logical block length, this capacity setting shall take effect when the new logical block length takes effect (i.e., after a successful FORMAT UNIT command);
c. If the NUMBER OF LOGICAL BLOCKS field is set to a value greater than the maximum capacity of the device and less than FFFFFFFF FFFFFFFFh, then the device server shall terminate the MODE SELECT 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 logical unit shall retain its previous block descriptor settings; or
d. If the NUMBER OF LOGICAL BLOCKS field is set to FFFFFFFF FFFFFFFFh, the logical unit shall be set to its maximum capacity. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length, the logical unit shall not become format corrupt. This capacity setting shall be retained through power cycles, hard resets, logical unit resets, and I_T nexus losses. If the content of the LOGICAL BLOCK LENGTH field is the same as the current logical block length this capacity setting shall take effect on successful completion of the MODE SELECT command. If the content of the LOGICAL BLOCK LENGTH field (i.e., new logical block length) is different than the current logical block length this capacity setting shall take effect when the new logical block length takes effect (i.e., after a successful FORMAT UNIT command).

The LOGICAL BLOCK LENGTH field specifies the length in bytes of each logical block. No change shall be made to any logical blocks on the medium until a format operation is initiated by an application client.

A device server shall respond to a MODE SENSE command (see 3.13 and 3.14) by reporting the length of the logical blocks as specified in the LOGICAL BLOCK LENGTH field sent in the last MODE SELECT command that contained a mode parameter block descriptor. If no MODE SELECT command with a block descriptor has been received then the current logical block length shall be returned (e.g., if the logical block length is 512 bytes and a MODE SELECT command occurs with the LOGICAL BLOCK LENGTH field set to 520 bytes, any MODE SENSE command would return 520 in the LOGICAL BLOCK LENGTH field). To determine the logical block length at which the logical unit is currently formatted, the application client shall use the READ CAPACITY command rather than the MODE SELECT command.

4.3.5 Mode page and subpage formats and page codes

The page_0 mode page format is defined in table 266.

Table 266. Page_0 mode page format

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0B)	PAGE CODE					
1	PAGE LENGTH (N-1)							
2	MODE PARAMETERS							
n								

The SUB_PAGE mode page format is defined in table 267.

Table 267. SUB_PAGE mode page format

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE					
1	SUBPAGE CODE							
2	(MSB)	PAGE LENGTH (N-3)						
3								(LSB)
4	MODE PARAMETERS							
n								

Each mode page contains a PS bit, an SPF bit, a PAGE CODE field, a PAGE LENGTH field, and a set of mode parameters. The page codes are defined in this subclause and in the mode parameter subclauses in the command standard for the specific device type. Each mode page with a SPF bit set to one contains a SUBPAGE CODE field.

SPF (SubPage Format) bit

0 A SubPage Format (SPF) bit set to zero indicates that the page_0 mode page format is being used.

1 A SPF bit set to one indicates that the SUB_PAGE mode page format is being used.

PS (parameters saveable) bit

1 When using the MODE SENSE command, a parameters saveable (PS) bit set to one indicates that the mode page may be saved by the logical unit in a nonvolatile, vendor specific location.

0 A PS bit set to zero indicates that the device server is not able to save the supported parameters. When using the MODE SELECT command, the PS bit is reserved.

PAGE CODE and SUBPAGE CODE fields

The PAGE CODE and SUBPAGE CODE fields identify the format and parameters defined for that mode page. Some page codes are defined as applying to all device types and other page codes are defined for the specific device type. The page codes that apply to a specific device type are defined in the command standard for that device type. The applicability of each subpage code matches that of the page code with which it is associated.

When using the MODE SENSE command, if page code 00h (vendor specific mode page) is implemented, the device server shall return that mode page last in response to a request to return all mode pages (page code 3Fh). When using the MODE SELECT command, this mode page should be sent last.

PAGE LENGTH field

The PAGE LENGTH field specifies the length in bytes of the mode parameters that follow. If the application client does not set this value to the value that is returned for the mode page by the MODE SENSE command, the 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 PARAMETER LIST. The logical unit may implement a mode page that is less than the full mode page length defined, provided no field is truncated and the PAGE LENGTH field correctly specifies the actual length implemented.

The mode parameters for each mode page are defined in the following subclauses, or in the mode parameters subclause in the command standard for the specific device type. Mode parameters not implemented by the logical unit shall be set to zero.

Table 268 defines the mode pages that are applicable to all device types that implement the MODE SELECT and MODE SENSE commands.

Table 268. Mode page codes and subpage codes

Page code	Subpage code	Mode Page Name	Reference
0Ah	00h	Control	4.3.8
0Ah	01h	Control Extension	4.3.9
02h	00h	Disconnect-Reconnect	4.3.10
15h	00h	Extended	
16h	00h	Extended Device-Type Specific	
03h	00h	Format Device mode page (Obsolete)	4.3.11
1Ch	00h	Informational Exceptions Control	4.3.12
09h	00h	Obsolete	
0Ch	00h	Notch (Obsolete)	4.3.13
1Ah	00h	Power Condition	4.3.14
18h	00h	Protocol Specific LUN	4.3.16
04h	00h	Rigid Drive Geometry Parameters page (Obsolete)	4.3.18
18h	01h - FEh	(See specific SCSI transport protocol)	
19h	00h	Protocol Specific Port	4.3.17
19h	01h - FEh	(See specific SCSI transport protocol)	
01h	00h - FEh	(See specific device type)	
04h - 08h	00h - FEh	(See specific device type)	
0Bh - 14h	00h - FEh	(See specific device type)	
1Bh	00h - FEh	(See specific device type)	
1Dh - 1Fh	00h - FEh	(See specific device type)	
20h - 3Eh	00h - FEh	(See specific device type)	
00h	not applicable	Vendor specific (does not require page format)	
3Fh	00h	Return all pages [a]	
3Fh	FFh	Return all pages and subpages [a]	
00h - 3Eh	FFh	Return all subpages [a]	
All page code and subpage code combinations not shown in this table are reserved.			
a Valid only for the MODE SENSE command			

4.3.6 Background Control mode page (1Ch)

The Background Control mode page (see table 269) is a subpage of the Informational Exception Control mode page (see 4.3.12) and provides controls over background operations. The mode page policy (see 4.4.10) for this subpage shall be shared.

Table 269. Background Control mode page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF(1b)	PAGE CODE (1Ch)					
1	SUBPAGE CODE (01h)							
2	(MSB)	PAGE LENGTH						
3								(LSB)
4	Reserved					S_L_FULL	LOWIR	EN_BMS
5	Reserved							EN_PS
6	(MSB)	BACKGROUND MEDIUM SCAN INTERVAL TIME						
7								(LSB)
8	(MSB)	BACKGROUND PRE-SCAN TIME LIMIT						
9								(LSB)
10	(MSB)	MINIMUM IDLE TIME BEFORE BACKGROUND SCAN						
11								(LSB)
12	(MSB)	MAXIMUM TIME TO SUSPEND BACKGROUND SCAN						
13								(LSB)
14	Reserved							
15								

The parameters saveable (PS) bit, the subpage format (SPF) bit, the PAGE CODE field, the SUBPAGE CODE field, and the PAGE LENGTH field are defined in 4.3.5.

S_L_FULL bit

0 A suspend on log full (S_L_FULL) bit set to zero specifies that the device server shall continue running a background scan operation (see SPC-4) even if the Background Scan Results log page (see 4.2.3) contains the maximum number of Background Scan log parameters (see table 237) supported by the logical unit.

1 A S_L_FULL bit set to one specifies that the device server shall suspend a background scan operation if the Background Scan Results log page contains the maximum number of Background scan log parameters supported by the logical unit.

LOWIR bit

0 A log only when intervention required (LOWIR) bit set to zero specifies that the device server shall log all suspected recoverable medium errors or unrecoverable medium errors that are identified during background scan operations in the Background Scan Results log page.

1 A LOWIR bit set to one specifies that the device server shall only log medium errors identified during back ground scan operations in the Background Scan Results log page that require application client intervention.

EN_BMS (Enable Background Medium Scan) bit

0 An enable background medium scan (EN_BMS) bit set to zero specifies that background medium scan is disabled.

1 An EN_BMS bit set to one specifies that background medium scan operations are enabled. If the EN_PS bit is also set to one then a background medium scan operation shall not start until after the pre-scan operation is halted or completed.

If a background medium scan operation is in progress when the EN_BMS bit is changed from one to zero, then the logical unit shall suspend the background medium scan operation before the device server completes the MODE SELECT command, and the background medium scan shall remain suspended until the EN_BMS bit is set to one, at which time the background medium scan operation shall resume, beginning with the logical block being tested when the background medium scan was suspended.

EN_PS (enable pre-scan) bit

0 An enable pre-scan (EN_PS) bit set to zero specifies that pre-scan is disabled. If a pre-scan operation is in progress when EN_PS is changed from a one to a zero then pre-scan is halted.

1 An EN_PS bit set to one specifies that a pre-scan operation is started after the next power on cycle.

Once this pre-scan has completed, another pre-scan shall not occur unless the EN_PS bit is set to zero, then set to one, and another power on cycle occurs.

BACKGROUND MEDIUM SCAN INTERVAL TIME field

The BACKGROUND MEDIUM SCAN INTERVAL TIME field specifies the minimum time, in hours, between the start of one pre-scan or background medium scan operation and the start of the next background medium scan operation. If the current background medium scan operation takes longer than the value specified in the BACKGROUND MEDIUM SCAN INTERVAL TIME field, then the logical unit shall:

a) continue the current background pre-scan operation or background medium scan operation until that background scan operation is complete; and
b) the logical unit shall start the next background medium scan operation starts upon completion of the current background pre-scan operation or background medium scan operation.

BACKGROUND PRE-SCAN TIME LIMIT field

The BACKGROUND PRE-SCAN TIME LIMIT field specifies the maximum time, in hours, for a background pre-scan operation to complete. If the background pre-scan operation does not complete within the specified time then the device server shall halt the background pre-scan operation. A value of zero specifies an unlimited timeout value.

MINIMUM IDLETIME BEFORE BACKGROUND SCAN field

The MINIMUM IDLE TIME BEFORE BACKGROUNG SCAN field specifies the time, in milliseconds, that the logical unit shall be idle after suspending a background scan operation before resuming a background scan operation (e.g., after the device server has completed all of the commands in the task set).

MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field

The MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field specifies the time, in milliseconds, that the device server should take to start processing a command received while a logical unit is performing a background scan operation.

4.3.7 Caching Parameters page (08h)

The Caching Parameters page for MODE SENSE/MODE SELECT defines the parameters that affect the use of the cache (see table 270).

Table 270. Caching Parameters page (08h)

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>PS</td><td>SPF (0b)</td><td colspan="6">PAGE CODE (08h)</td></tr>
<tr><td>1</td><td colspan="8">PAGE LENGTH (12h)</td></tr>
<tr><td>2</td><td>IC</td><td>ABPF</td><td>CAP</td><td>DISC</td><td>SIZE</td><td>WCE</td><td>MF</td><td>RCD</td></tr>
<tr><td>3</td><td colspan="4">DEMAND READ RETENTION PRIORITY</td><td colspan="4">WRITE RETENTION PRIORITY</td></tr>
<tr><td>4</td><td>(MSB)</td><td colspan="6" rowspan="2">DISABLE PREFETCH TRANSFER LENGTH</td><td></td></tr>
<tr><td>5</td><td></td><td>(LSB)</td></tr>
<tr><td>6</td><td>(MSB)</td><td colspan="6" rowspan="2">MINIMUM PREFETCH</td><td></td></tr>
<tr><td>7</td><td></td><td>(LSB)</td></tr>
<tr><td>8</td><td>(MSB)</td><td colspan="6" rowspan="2">MAXIMUM PREFETCH</td><td></td></tr>
<tr><td>9</td><td></td><td>(LSB)</td></tr>
<tr><td>10</td><td>(MSB)</td><td colspan="6" rowspan="2">MAXIMUM PREFETCH CEILING</td><td></td></tr>
<tr><td>11</td><td></td><td>(LSB)</td></tr>
<tr><td>12</td><td>FSW</td><td>LBCSS</td><td>DRA</td><td colspan="2">vendor specific</td><td colspan="2">Reserved</td><td>NV_DIS</td></tr>
<tr><td>13</td><td colspan="8">NUMBER OF CACHE SEGMENTS</td></tr>
<tr><td>14</td><td>(MSB)</td><td colspan="6" rowspan="2">CACHE SEGMENT SIZE</td><td></td></tr>
<tr><td>15</td><td></td><td>(LSB)</td></tr>
<tr><td>16</td><td colspan="8">Reserved</td></tr>
<tr><td>17
:
19</td><td>(MSB)</td><td colspan="6">Obsolete</td><td>(LSB)</td></tr>
</table>

PS (Parameter Savable) bit

The returned Parameter Savable (PS) bit of 1 indicates that page 08h parameter data is savable.

IC (Initiator Control) enable bit

1 When the Initiator Control (IC) enable bit is set to one, adaptive read look-ahead (ARLA) is disabled.

0 When IC is set to ZERO, ARLA is enabled. Since Seagate drives covered by this manual do not organize the cache according to size of segment, but rather by number of segments, this bit is used to enable or disable ARLA (in non-Seagate equipment, this might be used to designate cache size).

Note. ARLA cannot be disabled in some Seagate drives using the ASAII code. See individual drive's Product Manual, Volume 1.

ABPF (ABORT PREFETCH) bit

1 The ABORT PREFETCH (ABPF) bit, when set to one, with the DRA bit equal to zero, requests that the SCSI device abort the PREFETCH upon selection. The ABPF set to one takes precedence over the Minimum PREFETCH bytes.

0 When set to zero, with the DRA bit equal to zero, the termination of any active PREFETCH is dependent upon Caching Page bytes 4 through 11 and is operation and/or vendor-specific.

CAP (Caching Analysis Permitted) bit

1 The Caching Analysis Permitted (CAP) bit, when set to one, enables caching analysis.

0 A zero indicates caching analysis is disabled. Caching analysis results are placed in the SCSI logging information table, (see Table 241). See individual drive's Product Manual, Volume 1, SCSI Bus Conditions and Miscellaneous Features Supported table.

DISC (Discontinuity) bit

1 The Discontinuity (DISC) bit, when set to one, requests that the SCSI device continue the PREFETCH across time discontinuities, such as across cylinders or tracks up to the limits of the buffer, or segment, space available for PREFETCH.

0 When set to zero, the DISC requests that prefetches be truncated at time discontinuities.

SIZE (Size Enable) bit

1 The Size Enable (SIZE) bit, when set to one, indicates that the Cache Segment Size is to be used to control caching segmentation.

0 When SIZE equals zero, the Initiator requests that the Number of Cache Segments is to be used to control caching segmentation. For Seagate drives covered by this manual, SIZE is always zero.

WCE (Write Cache Enable) bit

0 SCSI WRITE commands may not return status and completion message bytes until all data has been written to the media.

1 SCSI WRITE commands may return status and completion message bytes as soon as all data has been received from the host.

MF (Multiplication Factor) bit

0 The Minimum PREFETCH and Maximum PREFETCH fields are interpreted as a number of logical blocks.

1 Specifies that the target shall interpret the minimum and maximum PREFETCH fields to be specified in terms of a scalar number which, when multiplied by the number of logical blocks to be transferred for the current command, yields the number of logical blocks for each of the respective types of PREFETCH.

RCD (READ Cache Disable) bit

0 SCSI READ commands may access the cache or the media.

1 SCSI READ commands must access the media. Data cannot come from the cache.

DEMAND READ RETENTION PRIORITY field

DEMAND READ RETENTION PRIORITY. The cache replacement algorithm does not distinguish between retention in the cache of host-requested data and PREFETCH data. Therefore, this half byte is always 0.

WRITE RETENTION PRIORITY field

WRITE RETENTION PRIORITY. The cache replacement algorithm does distinguish between retention in the cache of host-requested data and PREFETCH data. Therefore, this half byte is always 0.

DISABLE PREFETCH TRANSFER LENGTH field

DISABLE PREFETCH TRANSFER LENGTH. PREFETCH is disabled for any SCSI READ command whose requested transfer length exceeds this value.

MINIMUM PREFETCH field

The MINIMUM PREFETCH specifies the minimum number sectors to prefetch, regardless of the delay it may cause to other commands.

MAXIMUM PREFETCH field

The MAXIMUM PREFETCH specifies the maximum number of logical blocks that may be prefetched. The PREFETCH operation may be aborted before the MAXIMUM PREFETCH value is reached, but only if the MINIMUM PREFETCH value has been satisfied.

MAXIMUM PREFETCH CEILING field

The MAXIMUM PREFETCH Ceiling specifies an upper limit on the number of logical blocks computed as the maximum prefetch. If the MAXIMUM PREFETCH value is greater than the MAXIMUM PREFETCH CEILING, the value is truncated to the MAXIMUM PREFETCH CEILING value.

FSW (FORCE SEQUENTIAL WRITE) bit

1 The Force Sequential Write (FSW) bit, when set to one, indicates that multiple block writes are to be transferred over the SCSI bus and written to the media in an ascending, sequential, logical block order.

0 When the FSW bit equals zero, the target is allowed to reorder the sequence of writing addressed logical blocks in order to achieve a faster command completion.

The Logical Block Cache Segment Size (LBCSS) bit is not used at this time.

DRA (Disable READ-Ahead) bit

1 The Disable READ-Ahead (DRA) bit, when set to one, requests that the target not read into the buffer any logical blocks beyond the addressed logical block(s).

0 When the DRA bit equals zero, the target may continue to read logical blocks into the buffer beyond the addressed logical block(s).

NUMBER OF CACHE SEGMENTS byte

The NUMBER OF CACHE SEGMENTS byte gives the number of segments into which the host requests the drive divide the cache.

CACHE SEGMENT SIZE field

The CACHE SEGMENT SIZE field indicates the requested segment size in bytes. This manual assumes that the Cache Segment Size field is valid only when the Size bit is one.

DPO (DISABLE PAGE OUT) bit

If DPO = 1, the cache replacement algorithm should not replace existing cache data with the current data (if possible). If a cache segment must be overwritten, it should be made the LRU.

FUA (FORCE UNIT ACCESS) bit

READ command: If FUA = 1, the requested data must be read from the media. If cache data which overlaps the request has not yet been written to the media, it should be written before the read is allowed to occur.

WRITE command: If FUA = 1, all data must be written to the media before the SCSI operation returns the status and completion message bytes.

4.3.8 Control mode page (0Ah)

The Control mode page (see table 271) provides controls over SCSI features that are applicable to all device types (e.g., task set management and error logging). If a field in this mode page is changed while there is a task already in the task set, it is vendor specific whether the old or new value of the field applies to that task. The mode page policy (see 3.11) for this mode page shall be shared, or per I_T nexus.

Table 271. Control mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (0Ah)					
1	PAGE LENGTH (0Ah)							
2	TST			TMF_ONLY	Reserved	D_SENSE	GLTSD	RLEC
3	QUEUE ALGORITHM MODIFIER				Reserved	QERR		DQUE Obsolete
4	VS	RAC	UA_INTLCK_CTRL		SWP	RAERP Obsolete	UAAERP Obsolete	EAERP Obsolete
5	ATO	TAS	Reserved			AUTOLOAD MODE		
6	(MSB)	Obsolete formerly READY AER HOLDOFF PERIOD						
7								(LSB)
8	(MSB)	BUSY TIMEOUT PERIOD						
9								(LSB)
10	(MSB)	EXTENDED SELF-TEST COMPLETION TIME						
11								(LSB)

The PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

TST (task set type) field

A task set type (TST) field specifies the type of task set in the logical unit (see table 272).

Table 272. Task set type (TST) field

Code	Description
000b	The logical unit maintains one task set for all I_T nexuses
001b	The logical unit maintains separate task sets for each I_T nexus
010b - 111b	Reserved

Regardless of the mode page policy (see 4.4.10) for the Control mode page, the shared mode page policy shall be applied to the TST field. If the most recent MODE SELECT changes the setting of this field, then the device server shall establish a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus on which the MODE SELECT command was received, with the additional sense code set to MODE PARAMETERS CHANGED.

TMF_ONLY (Allow Task Management Functions Only) bit

0 The allow task management functions only (TMF_ONLY) bit set to zero specifies that the device server shall process tasks with the ACA task attribute received on the faulted I_T nexus when an ACA condition has been established (see SAM-4).

1 A TMF_ONLY bit set to one specifies that the device server shall terminate all tasks received on the faulted I_T nexus with an ACA ACTIVE status when an ACA condition has been established.

D_SENSE (DESCRIPTOR FORMAT SENSE DATA) bit

0 A descriptor format sense data (D_SENSE) bit set to zero specifies that the device server shall return the fixed format sense data when returning sense data in the same I_T_L_Q nexus transaction as a CHECK CONDITION status.

1 A D_SENSE bit set to one specifies that the device server shall return descriptor format sense data when returning sense data in the same I_T_L_Q nexus transaction as a CHECK CONDITION status, except as defined in 2.4.1.

GLTSD (Global Logging Target Save Disable) bit

0 A global logging target save disable (GLTSD) bit set to zero specifies that the logical unit implicitly saves, at vendor specific intervals, each log parameter in which the TSD bit (see 4.2) is set to zero.

1 A GLTSD bit set to one specifies that the logical unit shall not implicitly save any log parameters.

RLEC (Report Log Exception Condition) bit

1 A report log exception condition (RLEC) bit set to one specifies that the device server shall report log exception conditions as described in 4.2.1.

0 A RLEC bit set to zero specifies that the device server shall not report log exception conditions.

QUEUE ALGORITHM MODIFIER field

The QUEUE ALGORITHM MODIFIER field (see table 273) specifies restrictions on the algorithm used for reordering tasks having the SIMPLE task attribute (see SAM-4).

Table 273. QUEUE ALGORITHM MODIFIER field

Code	Description
0h	Restricted reordering
1h	Unrestricted reordering allowed
2h - 7h	Reserved
8h - Fh	Vendor specific

A value of zero in the QUEUE ALGORITHM MODIFIER field specifies that the device server shall order the processing sequence of tasks having the SIMPLE task attribute such that data integrity is maintained for that I_T nexus (i.e., if the transmission of new SCSI transport protocol requests is halted at any time, the final value of all data observable on the medium shall have exactly the same value as it would have if all the tasks had been given the ORDERED task attribute).

A value of one in the QUEUE ALGORITHM MODIFIER field specifies that the device server may reorder the processing sequence of tasks having the SIMPLE task attribute in any manner. Any data integrity exposures related to task sequence order shall be explicitly handled by the application client through the selection of appropriate commands and task attributes.

QERR (queue error management) field

The queue error management (QERR) field (see table 274) specifies how the device server shall handle other tasks when one task is terminated with CHECK CONDITION status (see SAM-4). The task set type (see the TST field definition in this subclause) defines which other tasks are affected. If the TST field equals 000b, then all tasks from all I_T nexuses are affected. If the TST field equals 001b, then only tasks from the same I_T nexus as the task that is terminated with CHECK CONDITION status are affected.

Table 274. Queue error management (QERR) field

Code	Definition
00b	If an ACA condition is established, the affected tasks in the task set shall resume after the ACA condition is cleared (see SAM-4). Otherwise, all tasks other than the task that received the CHECK CONDITION status shall be processed as if no error occurred.
01b	All the affected tasks in the task set shall be aborted when the CHECK CONDITION status is sent. If the TAS bit is set to zero, a unit attention condition (see SAM-4) shall be established for the initiator port associated with every I_T nexus that had tasks aborted except for the I_T nexus on which the CHECK CONDITION status was returned, with the additional sense code set to COMMANDS CLEARED BY ANOTHER INITIATOR. If the TAS bit is set to one, all affected tasks in the task set for I_T nexuses other than the I_T nexus for which the CHECK CONDITION status was sent shall be completed with a TASK ABORTED status and no unit attention shall be established. For the I_T nexus to which the CHECK CONDITION status is sent, no status shall be sent for the tasks that are aborted.
10b	Reserved
11b	Affected tasks in the task set belonging to the I_T nexus on which a CHECK CONDITION status is returned shall be aborted when the status is sent.

DQUE (Disable Queuing) bit

A Disable Queuing (DQUE) bit of one indicates that tagged queuing is disabled on the drive. Any pending commands in the queue for that I_T_X nexus is aborted. Any subsequent queue tag message received shall be rejected with a MESSAGE REJECT message and the I/O process shall be executed as an untagged command. A DQUE bit of zero indicates that tagged queuing is enabled, if the drive supports tagged Queuing.

Note. The DQUE bit has been declared obsolete by the T10 committee.

TAS (Task Aborted Status) bit

0 A task aborted status (TAS) bit set to zero specifies that aborted tasks shall be terminated by the device server without any response to the application client.

1 A TAS bit set to one specifies that tasks aborted by the actions of an I_T nexus other than the I_T nexus on which the command was received shall be terminated with a TASK ABORTED status (see SAM-4).

RAC (report a check) bit

1 The report a check (RAC) bit provides control of reporting long busy conditions or CHECK CONDITION status. A RAC bit set to one specifies that the device server should return CHECK CONDITION status rather than returning BUSY status if the reason for returning the BUSY status may persist for a longer time than that specified by the AUTOLOAD MODE field.

0 A RAC bit set to zero specifies that the device server may return BUSY status regardless of the length of time the reason for returning BUSY status may persist.

UA_INTLCK_CTRL (unit attention interlocks control) field

The unit attention interlocks control (UA_INTLCK_CTRL) field (see table 275) controls the clearing of unit attention conditions reported in the same I_T_L_Q nexus transaction as a CHECK CONDITION status and whether returning a status of BUSY, TASK SET FULL or RESERVATION CONFLICT results in the establishment of a unit attention condition (see SAM-4).

Table 275. Unit attention interlocks control (UA_INTLCK_CTRL) field

Code	Definition
00b	The logical unit shall clear any unit attention condition reported in the same I_T_L_Q nexus transaction as a CHECK CONDITION status and shall not establish a unit attention condition when a task is terminated with BUSY, TASK SET FULL, or RESERVATION CONFLICT status.
01b	Reserved [a]
10b [a]	The logical unit shall not clear any unit attention condition reported in the same I_T_L_Q nexus transaction as a CHECK CONDITION status and shall not establish a unit attention condition when a task is terminated with BUSY, TASK SET FULL, or RESERVATION CONFLICT status.
11b [a]	The logical unit shall not clear any unit attention condition reported in the same I_T_L_Q nexus transaction as a CHECK CONDITION status and shall establish a unit attention condition for the initiator port associated with the I_T nexus on which the BUSY, TASK SET FULL, or RESERVATION CONFLICT status is being returned. Depending on the status, the additional sense code shall be set to PREVIOUS BUSY STATUS, PREVIOUS TASK SET FULL STATUS, or PREVIOUS RESERVATION CONFLICT STATUS. Until it is cleared by a REQUEST SENSE command, a unit attention condition shall be established only once for a BUSY, TASK SET FULL, or RESERVATION CONFLICT status regardless to the number of commands terminated with one of those status values.

a A REQUEST SENSE command still clears any unit attention condition that it reports.

SWP (Software Write Protect) bit

1 A software write protect (SWP) bit set to one specifies that the logical unit shall inhibit writing to the medium after writing all cached or buffered write data, if any. When SWP is one, all commands requiring writes to the medium shall be terminated with CHECK CONDITION status, with the sense key set to DATA PROTECT, and the additional sense code set to WRITE PROTECTED. When SWP is one and the device type's command standard defines a write protect (WP) bit in the DEVICE-SPECIFIC PARAMETER field in the mode parameter header, the WP bit shall be set to one for subsequent MODE SENSE commands.

0 A SWP bit set to zero specifies that the logical unit may allow writing to the medium, depending on other write inhibit mechanisms implemented by the logical unit. When the SWP bit is set to zero, the value of the WP bit, if defined, is device type specific. For a list of commands affected by the SWP bit and details of the WP bit see the command standard for the specific device type.

The RAERP, UAAERP, and EAERP bits enable specific events to be reported via the asynchronous event reporting protocol. When all three bits are zero, the target shall not use asynchronous event reporting. AER is defined in SAM-2. These three bits have been declared obsolete by the T10 committee.

RAERP (Ready AER Permission) bit

1 A Ready AER Permission (RAERP) bit of one specifies that the target may issue an asynchronous event report upon completing its initialization sequence instead of generating a unit attention condition.

0 A RAERP bit of zero specifies that the target shall not issue an asynchronous event report upon completing its initialization sequence.

Note. If the target's default value for the RAERP bit is one and it does not implement saved parameters or include a hardware switch, then it may be impossible to disable the initialization sequence asynchronous event reporting.

UAAERP (UNIT ATTENTION AER Permission) bit

1 A UNIT ATTENTION AER Permission (UAAERP) bit of one specifies that the target may issue an asynchronous event report instead of creating a unit attention condition upon detecting an unit attention condition event (other than upon completing an initialization sequence).

0 A UAAERP bit of zero specifies that the target shall not issue an asynchronous event reporting instead of creating a unit attention condition.

EAERP (Error AER Permission) bit

1 An Error AER Permission (EAERP) bit of one specifies that the target may issue an asynchronous event report upon detecting a deferred error condition instead of waiting to report the deferred error on the next command.

0 An EAERP bit of zero specifies that the target shall not report deferred error conditions via an asynchronous event reporting.

ATO (application tag owner) bit

1 An application tag owner (ATO) bit set to one specifies that the contents of the LOGICAL BLOCK APPLICATION TAG field in the protection information, if any, shall not be modified by the device server. An ATO bit set to zero specifies that the contents of the LOGICAL BLOCK APPLICATION TAG field in the protection information, if any, may be modified by the device server.

0 If the ATO bit is set to zero, the device server shall ignore the contents of the LOGICAL BLOCK APPLICATION TAG field in the protection information when received from the application client.

AUTOLOAD MODE field

The AUTOLOAD MODE field specifies the action to be taken by a removable medium device server when a medium is inserted. For devices other than removable medium devices, this field is reserved. Table 276 shows the usage of the AUTOLOAD MODE field.

Table 276. AUTOLOAD MODE field

Code	Definition
000b	Medium shall be loaded for full access.
001b	Medium shall be loaded for medium auxiliary memory access only.
010b	Medium shall not be loaded.
011b - 111b	Reserved

READY AER HOLDOFF PERIOD field

The READY AER HOLDOFF PERIOD field specifies the minimum time in milliseconds after the target starts its initialization sequence that it shall delay before attempting to issue an asynchronous event report. This value may be rounded up.

Note. This field has been declared obsolete by the T10 committee.

BUSY TIMEOUT PERIOD field

The BUSY TIMEOUT PERIOD field specifies the maximum time, in 100 milliseconds increments, that the application client allows for the device server to return BUSY status for unanticipated conditions that are not a routine part of commands from the application client. This value may be rounded down as defined in Section 2.3. A 0000h value in this field is undefined by this manual. An FFFFh value in this field is defined as an unlimited period.

EXTENDED SELF-TEST COMPLETION TIME field

The EXTENDED SELF-TEST COMPLETION TIME field contains advisory data that is the time in seconds that the device server requires to complete an extended self-test when the device server is not interrupted by subsequent commands and no errors occur during processing of the self-test. The application client should expect this time to increase significantly if other commands are sent to the logical unit while a self-test is in progress or if errors occur during the processing of the self-test. Device servers supporting SELF-TEST CODE field values other than 000b for the SEND DIAGNOSTIC command (see 3.49) shall support the EXTENDED SELF-TEST COMPLETION TIME field. The EXTENDED SELF-TEST COMPLETION TIME field is not changeable.

Bits 0, 1, and 2 of byte 4 as well as bytes 6 and 7 provide controls for the obsolete asynchronous event reporting feature.

4.3.9 Control Extension mode page (0Ah)

The Control Extension mode page (see table 277) is a subpage of the Control mode page (see 4.3.8) and provides controls over SCSI features that are applicable to all device types. The mode page policy (see 4.4.10) for this mode page shall be shared. If a field in this mode page is changed while there is a task already in the task set, it is vendor specific whether the old or new value of the field applies to that task.

Table 277. Control Extension mode page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (0Ah)					
1	SUBPAGE CODE (01h)							
2	(MSB)	PAGE LENGTH (1Ch)						
3								(LSB)
4	Reserved					TCMOS	SCSIP	IALUAE
5	Reserved				INITIAL COMMAND PRIORITY			
6	Reserved							
31								

The PS bit, SPF bit, PAGE CODE field, SUBPAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

SCSIP (SCSI precedence) bit

1 A SCSI precedence (SCSIP) bit set to one specifies that the timestamp changed using a SET TIMESTAMP command shall take precedence over methods outside the scope of this manual.

0 A SCSIP bit set to zero specifies that methods outside this manual may change the timestamp and that the SET TIMESTAMP command is illegal.

TCMOS (Timestamp Changeable By Methods Outside this manual) bit

1 A timestamp changeable by methods outside this manual (TCMOS) bit set to one specifies that the timestamp may be initialized by methods outside the scope of this manual.

0 A TCMOS bit set to zero specifies that the timestamp shall not be changed by any method except those defined by this manual.

IALUAE (Implicit Asymmetric Logical Unit Access Enabled) bit

1 An Implicit Asymmetric Logical Unit Access Enabled (IALUAE) bit set to one specifies that implicit asymmetric logical unit access state changes are allowed.

0 An IALUAE bit set to zero specifies that implicit asymmetric logical unit access state changes be disallowed and indicates that implicit asymmetric logical unit access state changes are disallowed or not supported.

INITIAL COMMAND PRIORITY field

The INITIAL COMMAND PRIORITY field specifies the priority that may be used as the task priority (see SAM-4) for tasks received by the logical unit on any I_T nexus (i.e., on any I_T_L nexus) where a priority has not been modified by a SET PRIORITY command. If a MODE SELECT command specifies an INITIAL COMMAND PRIORITY value that is different than the current INITIAL PRIORITY, then the device server shall set any priorities that have not be set with a SET PRIORITY command to a value different than the new initial priority value to the new priority. The device server shall establish a unit attention condition for the initiator port associated with every I_T_L nexus that receives a new priority, with the additional sense code set to PRIORITY CHANGED.

4.3.10 Disconnect-Reconnect mode page (02h)

The Disconnect-Reconnect mode page (see table 278) provides the application client the means to tune the performance of the service delivery subsystem. The name for this mode page, disconnect-reconnect, comes from the SCSI parallel interface. The mode page policy (see 4.4.10) for this mode page shall be shared or per target port. If the SCSI target device contains more than one target port, the mode page policy should be per target port.

For Parallel SCSI, see Section 4.3.10 on page 326.

For Fibre Channel (FC), see Section 4.3.10.2 on page 330.

For Serial Attached SCSI (SAS), see Section 4.3.10.3 on page 333.

4.3.10.1 Disconnect-Reconnect mode page for Parallel SCSI

The Disconnect-Reconnect mode page controls parameters that affect one or more target ports. The parameters that may be implemented are specified in the SCSI transport protocol standard for the target port.

Table 278. Disconnect-Reconnect mode page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (02h)					
1	PAGE LENGTH (0Eh)							
2	BUFFER FULL RATIO							
3	BUFFER EMPTY RATIO							
4	(MSB)	BUS INACTIVITY LIMIT						
5								(LSB)
6	(MSB)	DISCONNECT TIME LIMIT						
7								(LSB)
8	(MSB)	CONNECT TIME LIMIT						
9								(LSB)
10	(MSB)	MAXIMUM BURST SIZE						
11								(LSB)
12	EMDP	FAIR ARBITRATION			DIMM	DTDC		
13	Reserved							
14	(MSB)	FIRST BURST SIZE						
15								(LSB)

The parameters for a target port affect its behavior regardless of which initiator port is forming an I_T nexus with the target port. The parameters may be accessed by MODE SENSE (see 3.13) and MODE SELECT (see 3.11) commands directed to any logical unit accessible through the target port. If a parameter value is changed, all the device servers for all logical units accessible through the target port shall establish a unit attention condition for the initiator port associated with every I_T nexus that includes the target port except the I_T nexus on which the MODE SELECT command was received, with the additional sense code set to MODE PARAMETERS CHANGED.

If a parameter that is not appropriate for the specific SCSI transport protocol implemented by the target port is non-zero, the 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 PARAMETER LIST.

An interconnect tenancy is a period of time during which a given pair of SCSI ports (i.e., an initiator port and a target port) are accessing the interconnect layer to communicate with each other (e.g., on arbitrated interconnects, a tenancy typically begins when a SCSI port successfully arbitrates for the interconnect and ends when the SCSI port releases the interconnect for use by other devices). Data and other information transfers take place during interconnect tenancies.

The PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

BUFFER FULL RATIO field

The BUFFER FULL RATIO field specifies to the target port how full the buffer should be during read operations prior to requesting an interconnect tenancy. Target ports that do not implement the requested ratio should round down to the nearest implemented ratio as defined in 2.3.

BUFFER EMPTY RATIO field

The BUFFER EMPTY RATIO field specifies to the target port how empty the buffer should be during write operations prior to requesting an interconnect tenancy. Target ports that do not implement the requested ratio should round down to the nearest implemented ratio as defined in 2.3.

The buffer full and buffer empty ratios are numerators of a fractional multiplier that has 256 as its denominator. A value of zero indicates that the target port determines when to request an interconnect tenancy consistent with the disconnect time limit parameter. These parameters are advisory to the target port.

Note. As an example, consider a target port with ten 512-byte buffers and a specified buffer full ratio of 3Fh. The formula is: INTEGER ((ratio/256)*number of buffers). Therefore in this example INTEGER((3Fh/256)*10) = 2. During the read operations described in this example, the target port should request an interconnect tenancy whenever two or more buffers are full.

BUS INACTIVITY LIMIT field

The BUS INACTIVITY LIMIT field specifies the maximum time that the target port is permitted to maintain an interconnect tenancy without data or information transfer. If the bus inactivity limit is exceeded, then the target port shall conclude the interconnect tenancy, within the restrictions placed on it by the applicable SCSI transport protocol. The contents of the DTDC field in this mode page also shall affect the duration of an interconnect tenancy. This value may be rounded as defined in 2.3. A value of zero specifies that there is no bus inactivity limit. Different SCSI transport protocols define different units of measure for the bus inactivity limit.

DISCONNECT TIME LIMIT field

The DISCONNECT TIME LIMIT field specifies the minimum time that the target port shall wait between interconnect tenancies. This value may be rounded. A value of zero specifies that there is no disconnect time limit. Different SCSI transport protocols define different units of measure for the disconnect time limit.

CONNECT TIME LIMIT field

The CONNECT TIME LIMIT field specifies the maximum duration of a single interconnect tenancy. If the connect time limit is exceeded, then the target port shall conclude the interconnect tenancy, within the restrictions placed on it by the applicable SCSI transport protocol. The contents of the DTDC field in this mode page also shall affect the duration of an interconnect tenancy. This value may be rounded. A value of zero specifies that there is no connect time limit. Different SCSI transport protocols define different units of measure for the connect time limit.

MAXIMUM BURST SIZE field

The MAXIMUM BURST SIZE field indicates the maximum amount of data that the target port shall transfer during a single data transfer operation. This value is expressed in increments of 512 bytes (i.e., a value of one means 512 bytes, two means 1 024 bytes, etc.). The relationship, if any, between data transfer operations and interconnect tenancies is defined in the individual SCSI transport protocol standards. A value of zero specifies there is no limit on the amount of data transferred per data transfer operation.

In terms of the SCSI transport protocol services, the device server shall limit the Request Byte Count argument to the Receive Data-Out protocol service and the Send Data-In protocol service to the amount specified in the MAXIMUM BURST SIZE field.

EMDP (Enable Modify Data Pointers) bit

The enable modify data pointers (EMDP) bit specifies whether or not the target port may transfer data out of order.

0 If the EMDP bit is set to zero, the target port shall not transfer data out of order.

1 If the EMDP bit is set to one, the target port is allowed to transfer data out of order.

FAIR ARBITRATION field

The FAIR ARBITRATION field specifies whether the target port should use fair or unfair arbitration when requesting an interconnect tenancy. The field may be used to specify different fairness methods as defined in the individual SCSI transport protocol standards.

DIMM (Disconnect Immediate) bit

0 A disconnect immediate (DIMM) bit set to zero specifies that the target port may transfer data for a command during the same interconnect tenancy in which it receives the command. Whether or not the target port does so may depend upon the target port's internal algorithms, the rules of the applicable SCSI transport protocol, and settings of the other parameters in this mode page.

1 A disconnect immediate (DIMM) bit set to one specifies that the target port shall not transfer data for a command during the same interconnect tenancy in which it receives the command.

DTDC (Data Transfer Disconnect Control) field

The data transfer disconnect control (DTDC) field (see table 279) defines other restrictions on when multiple interconnect tenancies are permitted. A non-zero value in the DTDC field shall take precedence over other interconnect tenancy controls represented by other fields in this mode page.

Table 279. Data transfer disconnect control

DTDC	Description
000b	Data transfer disconnect control is not used. Interconnect tenancies are controlled by other fields in this mode page.
001b	All data for a command shall be transferred within a single interconnect tenancy.
010b	Reserved
011b	All data and the response for a command shall be transferred within a single interconnect tenancy.
100b - 111b	Reserved

FIRST BURST SIZE field

The FIRST BURST SIZE field specifies the maximum amount of data that may be transferred to the target port for a command along with the command (i.e., the first burst). This value is expressed in increments of 512 bytes (i.e., a value of one means 512 bytes, two means 1 024 bytes, etc.). The meaning of a value of zero is SCSI transport protocol specific. SCSI transport protocols supporting this field shall provide an additional mechanism to enable and disable the first burst function.

In terms of the SCSI transport protocol services, the Receive Data-Out protocol service shall retrieve the first FIRST BURST SIZE amount of data from the first burst.

4.3.10.2 Disconnect-Reconnect mode page for FCP (02h)

4.3.10.2.1 Overview and format of Disconnect-Reconnect mode page for FCP

The Disconnect-Reconnect mode page for FCP (see Table 280) allows the application client to modify the behavior of the service delivery subsystem. This subclause specifies the parameters that are used by FCP devices and defines how FCP devices interpret the parameters. The application client communicates with the device server to determine what values are most appropriate for a device server. The device server communicates the parameter values in this mode page to the target FCP_Port, normally the Fibre Channel interface circuitry. This communication is internal to the SCSI target device and FCP device and is outside the scope of this manual. If a field or bit contains a value that is not supported by the FCP device, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

Table 280. Disconnect-Reconnect mode page (02h)

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	PAGE CODE (02h)					
1	PAGE LENGTH (0Eh)							
2	BUFFER FULL RATIO							
3	BUFFER EMPTY RATIO							
4	(MSB)	BUS INACTIVITY LIMIT						
5								(LSB)
6	(MSB)	DISCONNECT TIME LIMIT						
7								(LSB)
8	(MSB)	CONNECT TIME LIMIT						
9								(LSB)
10	(MSB)	MAXIMUM BURST SIZE						
11								(LSB)
12	EMDP	FAA	FAB	FAC	RESTRICTED	RESTRICTED		
13	Reserved							
14	(MSB)	FIRST BURST SIZE						
15								(LSB)

An interconnect tenancy is the period of time when an FCP device owns or may access a shared Fibre Channel interconnect. For arbitrated loops (see FC-AL-2) and Fibre Channel Class 1 connections, a tenancy typically begins when an FCP device successfully opens the connection and ends when the FCP device releases the connection for use by other device pairs. Data and other information transfers take place during interconnect tenancies.

Point-to-point or fabric-attached Class 2 or Class 3 links and many other configurations do not have a concept of interconnect tenancy and may perform transfers at any time.

BUFFER FULL RATIO field

The BUFFER FULL RATIO field indicates to the device server, during read operations, how full the buffer should be prior to requesting an interconnect tenancy. Device servers that do not implement the requested ratio should round down to the nearest implemented ratio as defined in 2.3. FCP devices attached to links that do not have the concept of interconnect tenancy shall round the ratio to zero and transmit data in a vendor specific manner.

The value contained in the BUFFER FULL RATIO field is defined by 4.3.10.1.

BUFFER EMPTY RATIO field

The BUFFER EMPTY RATIO field indicates to the device server, during write operations, how empty the buffer should be prior to transmitting an FCP_XFER_RDY IU that requests the initiator FCP_Port to send data. Device servers that do not implement the requested ratio should round down to the nearest implemented ratio as defined in 2.3.

The value contained in the BUFFER EMPTY RATIO field is defined by 4.3.10.1.

BUS INACTIVITY LIMIT field

The BUS INACTIVITY LIMIT field indicates the maximum time that the target FCP_Port is permitted to maintain an interconnect tenancy without data or information transfer, measured in transmission word increments. If the bus inactivity limit is exceeded or if the bus is inactive and the target FCP_Port holding the bus detects that the limit is going to be exceeded, the device server shall end the interconnect tenancy. This value may be rounded as defined in 2.3. A value of zero indicates that there is no bus inactivity limit.

Note. Because of the low overheads associated with initiating and closing bus tenancy on Fibre Channel links, device servers should end tenancies immediately upon completing the required transfers.

The BUS INACTIVITY LIMIT field is not applicable for FCP devices attached to links that do not have the concept of interconnect tenancy.

DISCONNECT TIME LIMIT field

The DISCONNECT TIME LIMIT field indicates the minimum delay between interconnect tenancies measured in increments of 128 transmission words. Target FCP_Ports in configurations having the concept of interconnect tenancy shall delay at least this time interval after each interconnect tenancy before beginning arbitration. The device server may round this value to any value it prefers. A value of zero indicates that the disconnect time limit does not apply.

The DISCONNECT TIME LIMIT field is not applicable for FCP devices attached to links that do not have the concept of interconnect tenancy.

CONNECT TIME LIMIT field

The CONNECT TIME LIMIT field indicates the maximum duration of a single interconnect tenancy, measured in increments of 128 transmission words. If the connect time limit is exceeded the device server shall conclude the interconnect tenancy, within the restrictions placed on it by the applicable Fibre Channel configuration. The device server may round this value to any value it prefers. A value of zero indicates that there is no connect time limit.

The CONNECT TIME LIMIT field is not applicable for FCP devices attached to links that do not have the concept of interconnect tenancy.

MAXIMUM BURST SIZE field

The MAXIMUM BURST SIZE field indicates the maximum size of all bytes in an FCP_DATA IU that the target FCP_Port shall transfer to the initiator FCP_Port in a single Data-In FCP_DATA IU or request from the initiator FCP_Port in an FCP_XFER_RDY IU. This parameter does not affect how much data is transferred in a single interconnect tenancy. This value is expressed in increments of 512 bytes (e.g., a value of 1 means 512 bytes, two means 1024 bytes, etc.). The device server may round this value down as defined in 2.3. A value of zero indicates there is no limit on the amount of data transferred per data transfer operation. This value shall be implemented by all FCP devices. The initiator FCP_Port and target FCP_Port may use the value of this parameter to adjust internal maximum buffering requirements.

EMDP bit

The enable modify data pointers (EMDP) bit indicates whether or not the target FCP_Port may use the random buffer access capability to reorder FCP_DATA IUs for a single SCSI command. If the EMDP bit is set to zero, the target FCP_Port shall generate continuously increasing relative offset values for each FCP_DATA IU for a single SCSI command.

1 If the EMDP bit is set to one, the target FCP_Port may transfer the FCP_DATA IUs for a single SCSI command in any order.

0 If the EMDP bit is set to zero, data overlay is prohibited even if it is allowed by the state of the PRLI FCP Service Parameter page DATA OVERLAY ALLOWED bit.

The EMDP bit does not affect the order of frames within a Sequence. The enable modify data pointers function is optional for all FCP devices.

For bidirectional commands, the EMDP bit applies independently to the read operation and write operation. If the EMDP bit is set to zero, the target FCP_Port shall generate continuously increasing relative offset values for the read operation and the write operation, but there is no read operation to write operation or write operation to read operation ordering requirement.

FAA, FAB, FAC bits

The fairness access (FA) bits, FAA, FAB, and FAC, indicate whether a target FCP_Port attached to an arbitrated loop (see FC-AL-2) shall use the access fairness algorithm when beginning the interconnect tenancy.

An FA bit set to one indicates that the target FCP_Port shall use the access fairness algorithm for the specified frames. An FA bit set to zero indicates that the target FCP_Port may choose to not use the access fairness algorithm. The FAA bit controls arbitration when the target FCP_Port has one or more FCP_DATA IU frames to send to an initiator FCP_Port.

The FAB bit controls arbitration when the target FCP_Port has one or more FCP_XFER_RDY IU frames to send to an initiator FCP_Port.

The FAC bit controls arbitration when the target FCP_Port has an FCP_RSP IU frame to send to an initiator FCP_Port. If the target FCP_Port intends to send multiple frame types, it may choose to not use the access fairness algorithm if any applicable FA bit is set to zero. FCP devices attached to links that do not have the concept of interconnect tenancy shall ignore the FA bits. The FA bits are optional for all FCP devices.

FIRST BURST SIZE field

When the WRITE FCP_XFER_RDY DISABLED bit is negotiated as being set to one in the PRLI FCP Service Parameter page, the FIRST BURST SIZE field indicates the maximum amount of all bytes that shall be transmitted in the first FCP_DATA IU sent from the initiator FCP_Port to the target FCP_Port. If all data is transmitted in the first IU, no subsequent FCP_XFER_RDY IUs shall be transmitted by the target FCP_Port. If the maximum amount of data has been transmitted, but more data remains to be transferred, the target FCP_Port shall request that data with subsequent FCP_XFER_RDY IUs.

When the WRITE FCP_XFER_RDY DISABLED bit is negotiated as being set to zero in the PRLI FCP Service Parameter page, the FIRST BURST SIZE field is ignored and permission to transmit data from the initiator FCP_Port to the target FCP_Port is managed using FCP_XFER_RDY IUs. For data transmissions from the target FCP_Port to the initiator FCP_Port, the FIRST BURST SIZE field is ignored.

The FIRST BURST SIZE field value is expressed in increments of 512 bytes (e.g., a value of one means 512 bytes, two means 1024 bytes). A value of zero indicates that there is no first burst size limit. The FIRST BURST SIZE field shall be implemented by all FCP devices that support the WRITE FCP_XFER_RDY DISABLED bit being set to one. The application client and device server may use the value of this parameter to adjust internal maximum buffering requirements.

4.3.10.3 Disconnect-Reconnect mode page for SAS (02h)

4.3.10.3.1 Disconnect-Reconnect mode page overview

The Disconnect-Reconnect mode page for SAS provides the application client the means to tune the performance of the service delivery subsystem. Table 281 defines the parameters which are applicable to SSP. If any field in the Disconnect-Reconnect mode page is not implemented, the value assumed for the functionality of the field shall be zero (i.e., as if the field in the mode page is implemented and the field is set to zero).

The application client sends the values in the fields to be used by the device server to control the SSP connections by means of a MODE SELECT command. The device server shall then communicate the field values to the SSP target port. The field values are communicated from the device server to the SSP target port in a vendor-specific manner.

SAS devices shall only use the parameter fields defined below in Table 281. If any other fields within the Disconnect-Reconnect mode page of the MODE SELECT command contain a non-zero value, the device server shall terminate the MODE SELECT 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.

Table 281. Disconnect-Reconnect mode page for SAS

<table>
<tr><th>Bit
Byte</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>0</td><td>PS</td><td>RESERVED</td><td colspan="6">PAGE CODE (02h)</td></tr>
<tr><td>1</td><td colspan="8">PAGE LENGTH (0Eh)</td></tr>
<tr><td>2</td><td colspan="8">RESERVED</td></tr>
<tr><td>3</td><td colspan="8">RESERVED</td></tr>
<tr><td>4</td><td colspan="8">(MSB)
BUS INACTIVITY TIME LIMIT</td></tr>
<tr><td>5</td><td colspan="8">(LSB)</td></tr>
<tr><td>6</td><td colspan="8">RESERVED</td></tr>
<tr><td>7</td><td colspan="8"></td></tr>
<tr><td>8</td><td colspan="8">(MSB)
MAXIMUM CONNECT TIME LIMIT</td></tr>
<tr><td>9</td><td colspan="8">(LSB)</td></tr>
<tr><td>10</td><td colspan="8">(MSB)
MAXIMUM BURST SIZE</td></tr>
<tr><td>11</td><td colspan="8">(LSB)</td></tr>
<tr><td>12</td><td colspan="8">RESERVED</td></tr>
<tr><td>13</td><td colspan="8">RESERVED</td></tr>
<tr><td>14</td><td colspan="8">(MSB)
FIRST BURST SIZE</td></tr>
<tr><td>15</td><td colspan="8">(LSB)</td></tr>
</table>

The PARAMETERS SAVEABLE (PS) bit is defined in 3.11.

The PAGE CODE (PS) field shall be set to 02h.

The PAGE LENGTH field shall be set to 0Eh.

BUS INACTIVITY TIME LIMIT field

The value in the BUS INACTIVITY TIME LIMIT field contains the maximum period that an SSP target port is permitted to maintain a connection without transferring a frame to the SSP initiator port. This value shall be the number of 100 µs increments between frames that the SSP target port transmits during a connection. When this number is exceeded, the SSP target port shall prepare to close the connection (i.e., by requesting to have the link layer transmit DONE). This value may be rounded as defined in 2.3. A value of zero in this field shall specify that there is no bus inactivity time limit.

MAXIMUM CONNECT TIME LIMIT field

The value in the MAXIMUM CONNECT TIME LIMIT field contains the maximum duration of a connection. This value shall be the number of 100 µs increments that an SSP target port transmits during a connection after which the SSP target port shall prepare to close the connection (e.g., a value of one in this field means that the time is less than or equal to 100 µs and a value of two in this field means that the time is less than or equal to 200 µs). If an SSP target port is transferring a frame when the maximum connection time limit is exceeded, the SSP target port shall complete transfer of the frame before preparing to close the connection. A value of zero in this field shall specify that there is no maximum connection time limit. The maximum connection time limit is enforced by the port layer. This value may be rounded as defined in 2.3.

MAXIMUM BURST SIZE field

For read data, the value in the MAXIMUM BURST SIZE field contains the maximum amount of data that is transferred during a connection by an SSP target port per I_T_L_Q nexus without transferring at least one frame for a different I_T_L_Q nexus. If the SSP target port:

a) has read data to transfer for only one I_T_L_Q nexus, and

b) has no requests to transfer write data for any I_T_L_Q nexus;

then the SSP target port shall prepare to close the connection after the amount of data specified by the MAXIMUM BURST SIZE field is transferred to the SSP initiator port.

For write data, the value shall specify the maximum amount of data that an SSP target port requests via a single XFER_RDY frame.

This value shall be specified in 512-byte increments (e.g., a value of one in this field means that the number of bytes transferred to the SSP initiator port for the nexus is less than or equal to 512 and a value of two in this field means that the number of bytes transferred to the SSP initiator port for the nexus is less than or equal to 1 024). The device server may round this value down as defined in 2.3. A value of zero in this field shall specify that there is no maximum burst size.

In terms of the SCSI transport protocol services, the device server shall limit the Request Byte Count argument to the Receive Data-Out () protocol service and the Send Data-In () protocol service to the amount specified in this field.

FIRST BURST SIZE field

If the ENABLE FIRST BURST field in the COMMAND frame is set to zero, the FIRST BURST SIZE field is ignored.

If the ENABLE FIRST BURST field in the COMMAND frame is set to one, the value in the FIRST BURST SIZE field contains the maximum amount of write data in 512-byte increments that may be sent by the SSP initiator port to the SSP target port without having to receive an XFER_RDY frame from the SSP target port (e.g., a value of one in this field means that the number of bytes transferred by the SSP initiator port is less than or equal to 512 and a value of two in this field means that the number of bytes transferred by the SSP initiator port is less than or equal to 1 024).

Specifying a non-zero value in the FIRST BURST SIZE field is equivalent to an implicit XFER_RDY frame for each command requiring write data where the WRITE DATA LENGTH field of the XFER_RDY frame is set to 512 times the value of the FIRST BURST SIZE field

The rules for data transferred using the value in the FIRST BURST SIZE field are the same as those used for data transferred for an XFER_RDY frame (i.e., the number of bytes transferred using the value in the FIRST BURST SIZE field is as if that number of bytes was requested by an XFER_RDY frame).

If the amount of data to be transferred for the command is less than the amount of data specified by the FIRST BURST SIZE field, the SSP target port shall not transmit an XFER_RDY frame for the command. If the amount of data to be transferred for the command is greater than the amount of data specified by the FIRST BURST SIZE field, the SSP target port shall transmit an XFER_RDY frame after it has received all of the data specified by the FIRST BURST SIZE field from the SSP initiator port. All data for the command is not required to be transferred during the same connection in which the command is transferred.

A value of zero in this field shall specify that there is no first burst size (i.e., an SSP initiator port shall transmit no write DATA frames to the SSP target port before receiving an XFER_RDY frame).

4.3.11 Format Parameters page (03h)

This mode page has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

The Format Parameters page implementation is defined in Table 282. This table summarizes the function and defines the default or changeability status for each bit. The actual implementation of reserving spare areas for defect management takes place during the FORMAT UNIT command.

Note. In Table 282, zone refers to defect management zone (one or more tracks), not a ZBR (variable track capacity recording) zone.ZBR zones are referred to as notches (page 0Ch is the Notch page).

Table 282. Format Parameters page (03h)

Bit / Byte	7	6	5	4	3	2	1	0
Page Descriptor Header								
0	PS	Reserved	PAGE CODE (03h)					
1	PAGE LENGTH (16h)							
Format Parameters								
2,3	TRACKS PER ZONE							
4,5	ALTERNATE SECTORS PER ZONE							
6,7	ALTERNATE TRACKS PER ZONE							
8,9	ALTERNATE TRACKS PER VOLUME							
10,11	SECTORS PER TRACK							
12,13	DATA BYTES PER PHYSICAL SECTOR							
14,15	INTERLEAVE							
16,17	TRACK SKEW FACTOR							
18,19	CYLINDER SKEW FACTOR							
20	SSEC	HSEC	RMB	SURF	DRIVE TYPE			
21,22,23	Reserved							

PS (Page Savable) bit

The only time this page of parameters may be sent is immediately before sending a FORMAT UNIT command to the drive. The Current parameters for this page are updated immediately but any changes between these Current parameters and the existing media format are not in effect until after the FORMAT UNIT command is completed. A PS bit of 1 indicates this page is savable. The PS bit is not used with the MODE SELECT command.

TRACKS PER ZONE field

The TRACKS PER ZONE field indicates the number of tracks the drive allocates to each defect management zone. A zone can be one or more tracks or one or more cylinders. See individual drive's Product Manual, Volume 1, for number of tracks allocated to each defect management zone for that drive model.

ALTERNATE SECTORS PER ZONE field

The ALTERNATE SECTORS PER ZONE field indicates the number of spare sectors to be reserved for the defined defect management zone. A value of zero indicates that no sectors are to be reserved in each zone for defect management. This is to accommodate hosts that want to manage the defects themselves.

ALTERNATE TRACKS PER ZONE field

The ALTERNATE TRACKS PER ZONE field indicates the number of spare tracks to be reserved at the end of each defect management zone. A value of zero indicates that no spare tracks are to be reserved in each zone for defect management by the drive.

ALTERNATE TRACKS PER VOLUME field

The ALTERNATE TRACKS PER VOLUME field indicates the number of spare tracks to be reserved at the end of the drive volume. The drive uses these locations for replacing defective sectors. A value of zero indicates that no spare tracks are to be reserved at the end of the unit for defect management. The initiator may change this value for a number between 0 and 255 that is a multiple of the total number of Data Read/Write heads installed. However, it is not changeable on some products.

SECTORS PER TRACK field

The SECTORS PER TRACK field indicates the average number of physical sectors the drive has per disc track. This value depends on the selected sector size and ZBR zones. The number of user accessible sectors per track may be fewer than the reported value, since sectors per Track includes sectors set aside for defect management. This value cannot be used to calculate drive user accessible capacity.

Note. The value cannot be directly selected with the MODE SELECT command, but is a report of how the drive is configured.

DATA BYTES PER PHYSICAL SECTOR field

The DATA BYTES PER PHYSICAL SECTOR field indicates the number of data bytes the drive shall allocate per physical sector. This value equals the block length reported in the MODE SENSE block descriptor. The bytes per physical sector is not directly changeable by the initiator and is not verified on a MODE SELECT command.

INTERLEAVE field

The INTERLEAVE field is the interleave value sent to the drive during the last FORMAT UNIT command.

Note. This field is valid only for MODE SENSE commands. The drive ignores this field during MODE SELECT commands.

TRACK SKEW FACTOR field

The TRACK SKEW FACTOR field indicates the average number of physical sectors between the last logical block on one track and the first logical block on the next sequential track of the same cylinder. A value of zero indicates no skew.

Note. This value is not changeable by a SCSI initiator port.

CYLINDER SKEW FACTOR field

The CYLINDER SKEW FACTOR field indicates the average number of physical sectors between the last logical block of one cylinder and the first logical block of the next cylinder. A value of zero indicates no skew. Cylinder skew will be utilized by a drive but is not changeable by a SCSI initiator port.

DRIVE TYPE field

The DRIVE TYPE field bits are defined as follows:

a) The Hard Sectoring (HSEC) bit (bit 6) set to one indicates the drive shall use hard sector formatting.

b) Bits 0-5, and 7 are not implemented by the drive and are always zero. All bits (0-7) are not changeable.

c) See individual drive’s Product Manual, Volume 1, MODE SENSE Data clause for changeable values.

4.3.12 Informational Exceptions Control mode page (1Ch)

The Informational Exceptions Control mode page (see table 283) defines the methods used by the device server to control the reporting and the operations of specific informational exception conditions. This page shall only apply to informational exceptions that report an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED or an additional sense code of WARNING to the application client. The mode page policy (see 4.4.10) for this mode page shall be shared, or per I_T nexus.

Informational exception conditions occur as the result of vendor specific events within a logical unit. An informational exception condition may occur asynchronous to any commands issued by an application client.

Note. Storage devices that support SMART (Self-Monitoring Analysis and Reporting Technology) for predictive failure software should use informational exception conditions.

Table 283. Informational Exceptions Control mode page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (1Ch)					
1	PAGE LENGTH (0Ah)							
2	PERF	Reserved	EBF	EWASC	DEXCPT	TEST	EBACKERR	LOGERR
3	Reserved				MRIE			
4	(MSB)	INTERVAL TIMER						
7								(LSB)
8	(MSB)	REPORT COUNT						
11								(LSB)

PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field

The PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

LOGERR (Log Error) bit

0 If the log errors (LOGERR) bit is set to zero, the logging of informational exception conditions by a device server is vendor specific.

1 If the LOGERR bit is set to one, the device server shall log informational exception conditions.

TEST bit

1 A TEST bit set to one shall create a test device failure at the next interval time, as specified by the INTERVAL TIMER field, if the DEXCPT bit is set to zero. When the TEST bit is set to one, the MRIE and REPORT COUNT fields shall apply as if the TEST bit were zero. The test device failure shall be reported with the additional sense code set to FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE). If both the TEST bit and the DEXCPT bit are one, the MODE SELECT 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 PARAMETER LIST.

0 A TEST bit set to zero shall instruct the device server not to generate any test device failure notifications.

EBACKERR (enable background error) bit

- **0** An enable background error (EBACKERR) bit set to zero indicates the target shall disable reporting of background self-test errors (SPC-4) and background scan errors (see SBC-3).
- **1** An EBACKERR bit set to one indicates reporting of background self-test errors and background scan errors shall be enabled. The method for reporting background self-test errors and background scan errors is determined by contents of the mrie field. Background self-test errors and background scan errors shall be reported as soon as the method specified in the mrie field occurs (i.e., the interval timer field and report count field do not apply for background self-test errors and background scan errors)

DEXCPT (Disable Exception Control) bit

- **0** A disable exception control (DEXCPT) bit set to zero indicates the failure prediction threshold exceeded reporting shall be enabled. The method for reporting the failure prediction threshold exceeded when the DEXCPT bit is set to zero is determined from the MRIE field.
- **1** A DEXCPT bit set to one indicates the device server shall disable reporting of the failure prediction threshold exceeded. The MRIE field is ignored when DEXCPT is set to one and EWASC is set to zero.

EWASC (Enable Warning) bit

- **0** If the enable warning (EWASC) bit is set to zero, the device server shall disable reporting of the warning. The MRIE field is ignored when DEXCPT is set to one and EWASC is set to zero.
- **1** If the EWASC bit is set to one, warning reporting shall be enabled. The method for reporting the warning when the EWASC bit is set to one is determined from the MRIE field.

EBF (Enable Background Function) bit

- **1** If background functions are supported and the Enable Background Function (EBF) bit is set to one, then the device server shall enable background functions.
- **0** If the EBF bit is set to zero, the device server shall disable the functions. Background functions with separate enable control bits (e.g., background medium scan defined in SBC-3) are not controlled by this bit.

For the purposes of the EBF bit, background functions are defined as idle time functions that may impact performance that are performed by a device server operating without errors but do not impact the reliability of the logical unit (e.g., read scan).

PERF (Performance) bit

- **0** If the performance (PERF) bit is set to zero, informational exception operations that are the cause of delays are acceptable.
- **1** If the PERF bit is set to one, the device server shall not cause delays while doing informational exception operations. A PERF bit set to one may cause the device server to disable some or all of the informational exceptions operations, thereby limiting the reporting of informational exception conditions.

MRIE (Method Of Reporting Informational Exceptions) field

The value in the method of reporting informational exceptions (MRIE) field (see table 284) defines the method that shall be used by the device server to report informational exception conditions. The priority of reporting multiple information exceptions is vendor specific.

Table 284. Method of reporting informational exceptions (MRIE) field (Sheet 1 of 2)

MRIE	Description
0h	**No reporting of informational exception condition:** The device server shall not report information exception conditions.
1h	**Asynchronous event reporting:** Obsolete
2h	**Generate unit attention:** The device server shall report informational exception conditions by establishing a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus, with the additional sense code set to indicate the cause of the informational exception condition. As defined in SAM-4, the command that has the CHECK CONDITION status with the sense key set to UNIT ATTENTION is not processed before the informational exception condition is reported.
3h	**Conditionally generate recovered error:** The device server shall report informational exception conditions, if the reporting of recovered errors is allowed, [a] by returning a CHECK CONDITION status. If the TEST bit is set to zero, the status may be returned after the informational exception condition occurs on any command for which GOOD status or INTERMEDIATE status would have been returned. If the TEST bit is set to one, the status shall be returned on the next command received on any I_T nexus that is normally capable of returning an informational exception condition when the test bit is set to zero. The sense key shall be set to RECOVERED ERROR and the additional sense code shall indicate the cause of the informational exception condition. The command that returns the CHECK CONDITION for the informational exception shall complete without error before any informational exception condition may be reported.
4h	**Unconditionally generate recovered error:** The device server shall report informational exception conditions, regardless of whether the reporting of recovered errors is allowed,[a] by returning a CHECK CONDITION status. If the TEST bit is set to zero, the status may be returned after the informational exception condition occurs on any command for which GOOD status or INTERMEDIATE status would have been returned. If the TEST bit is set to one, the status shall be returned on the next command received on any I_T nexus that is normally capable of returning an informational exception condition when the TEST bit is set to zero. The sense key shall be set to RECOVERED ERROR and the additional sense code shall indicate the cause of the informational exception condition. The command that returns the CHECK CONDITION for the informational exception shall complete without error before any informational exception condition may be reported.
5h	**Generate no sense:** The device server shall report informational exception conditions by returning a CHECK CONDITION status. If the TEST bit is set to zero, the status may be returned after the informational exception condition occurs on any command for which GOOD status or INTERMEDIATE status would have been returned. If the TEST bit is set to one, the status shall be returned on the next command received on any I_T nexus that is normally capable of returning an informational exception condition when the TEST bit is set to zero. The sense key shall be set to NO SENSE and the additional sense code shall indicate the cause of the informational exception condition. The command that returns the CHECK CONDITION for the informational exception shall complete without error before any informational exception condition may be reported.
a Error reporting is controlled by the post error (PER) bit in the Read-Write Error Recovery mode page (see 4.3.15).	

Table 284. Method of reporting informational exceptions (MRIE) field (Sheet 2 of 2)

MRIE	Description
6h	**Only report informational exception condition on request:** The device server shall preserve the informational exception(s) information. To find out about information exception conditions the application client polls the device server by issuing a REQUEST SENSE command. In the REQUEST SENSE parameter data that contains the sense data, the sense key shall be set to NO SENSE and the additional sense code shall indicate the cause of the informational exception condition.
7h - Bh	Reserved
Ch - Fh	Vendor specific
a Error reporting is controlled by the post error (PER) bit in the Read-Write Error Recovery mode page (see 4.3.15).	

INTERVAL TIMER field

The value in the INTERVAL TIMER field is the period in 100 millisecond increments for reporting that an informational exception condition has occurred. The device server shall not report informational exception conditions more frequently than the time specified by the INTERVAL TIMER field and shall report them after the time specified by INTERVAL TIMER field has elapsed. After the informational exception condition has been reported the interval timer shall be restarted. A value of zero or FFFF FFFF FFFF FFFFh in the INTERVAL TIMER field indicates that the period for reporting an informational exception condition is vendor specific.

The maintaining of the interval timer and the report counter across power cycles, hard resets, logical unit resets, and I_T nexus losses is vendor specific.

REPORT COUNT field

The REPORT COUNT field specifies the maximum number of times the device server may report an informational exception condition to the application client. A value of zero in the REPORT COUNT field indicates there is no limit on the number of times the device server reports an informational exception condition.

4.3.13 Notch page (0Ch)

This mode page has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

The Notch page (Table 285) contains parameters for direct access devices that implement a variable number of blocks per cylinder and support this page. Each section of the drive with a different number of blocks per cylinder is referred to as a notch.

Table 285. Notch page (0Ch)

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	PAGE CODE (0Ch)					
1	PAGE LENGTH (16h)							
2	ND	LPN	Reserved					
3	Reserved							
4,5	(MSB)	MAXIMUM NUMBER OF NOTCHES						
								(LSB)
6,7	(MSB)	ACTIVE NOTCH						
								(LSB)
8-11	(MSB)	STARTING BOUNDARY						
								(LSB)
12-15	(MSB)	ENDING BOUNDARY						
								(LSB)
16-23	(MSB)	PAGES NOTCHED						
								(LSB)

PS (Parameter Savable) bit

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

ND (Notched Drive) bit

0 A Notched Drive (ND) bit of zero indicates that the device is not notched and that all other parameters in this page shall be returned as zero by the drive.

1 An ND bit of one indicates that the drive is notched. For each supported active notch value, this page defines the starting and ending boundaries of the notch.

LPN (Logical or Physical Notch) bit

0 A Logical or Physical Notch (LPN) bit of zero indicates that the notch boundaries are based on the physical parameters of the drive. The cylinder is considered most significant, the head least significant.

1 An LPN bit of one indicates that the notch boundaries are based on logical blocks on the drive.

MAXIMUM NUMBER OF NOTCHES field

The MAXIMUM NUMBER OF NOTCHES field indicates the maximum number of notches supported by the drive.

ACTIVE NOTCH field

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

STARTING BOUNDARY field

The STARTING BOUNDARY field indicates the beginning of the active notch or, if the active notch is zero, the beginning boundary of the drive. If the LPN bit is one, then the four bytes represent a LOGICAL BLOCK ADDRESS. If the LPN bit is zero, then the three most significant bytes shall represent the cylinder number and the least significant byte shall represent the head number. When used with the MODE SELECT command, this field is ignored.

ENDING BOUNDARY field

The ENDING BOUNDARY field indicates the ending of the active notch or, if the active notch is zero, the ending of the drive. If the LPN bit is one, then the four bytes represent LOGICAL BLOCK ADDRESS. If the LPN bit is zero, then the three most significant bytes shall represent the cylinder number and the least significant byte shall represent the head number. When used with the MODE SELECT command, this field is ignored.

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

PAGES NOTCHED field

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

See individual drive’s Product Manual, Volume 1, MODE SENSE Data clause, for a table showing codes that indicate which of these bits are changeable by the host using the MODE SELECT command.

4.3.14 Power Condition mode page (1Ah)

The Power Condition mode page provides an application client with methods to control the power condition of a logical unit (see 3.12). These methods include:

a) Specifying that the logical unit transition to a power condition without delay; and

b) Activating and setting of idle condition and standby condition timers to specify that the logical unit wait for a period of inactivity before transitioning to a specified power condition.

The mode page policy (see 3.11) for this mode page shall be shared.

When a device server receives a command while in a power condition based on a setting in the Power Condition mode page, the logical unit shall transition to the power condition that allows the command to be processed. If either the idle condition timer or the standby condition timer has been set, then they shall be reset on receipt of the command. On completion of the command, the timer(s) shall be started.

Logical units that contain cache memory shall write all cached data to the medium for the logical unit (e.g.,as a logical unit does in response to a SYNCHRONIZE CACHE command as described in 3.53 and 3.54) prior to entering into any power condition that prevents accessing the media (e.g., before a hard drive stops its spindle motor during transition to the standby power condition).

The logical unit shall use the values in the Power Condition mode page to control its power condition after a power on or a hard reset until a START STOP UNIT command setting a power condition is received.

Table 286 defines the Power Condition mode page.

Table 286. Power Condition mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0B)	PAGE CODE (1Ah)					
1	PAGE LENGTH (0Ah)							
2	Reserved							STANDBY_Y
3	Reserved				IDLE_C	IDLE_B	IDLE_A	STANDBY_Z
4	(MSB)	IDLE_A CONDITION TIMER						
7								(LSB)
8	(MSB)	STANDBY_Z CONDITION TIMER						
11								(LSB)
	(MSB)	IDLE_B CONDITION TIMER						
								(LSB)
	(MSB)	IDLE_C CONDITION TIMER						
								(LSB)
	(MSB)	STANDBY_Y CONDITION TIMER						
								(LSB)
	(MSB)	Reserved						
								(LSB)

PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field

The PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

The behavior of the idle and standby condition timers controlled by this mode page is defined in the power condition overview and the power condition state machine (see SPC-4).

STANDBY_Y bit

1 If the STANDBY_Y bit is set to one, then the standby_y condition timer is enabled.

0 If the STANDBY_Y bit is set to zero, then the device server shall ignore the standby_y condition timer.

IDLE_C bit

1 If the IDLE_C bit is set to one, then the idle_c condition timer is enabled.

0 If the IDLE_C bit is set to zero, then the device server shall ignore the idle_c condition timer.

IDLE_B bit

1 If the IDLE_B bit is set to one, then the idle_b condition timer is enabled.

0 If the IDLE_B bit is set to zero, then the device server shall ignore the idle_b condition timer.

IDLE_A bit

1 If the IDLE_A bit is set to one, then the idle_a condition timer is enabled.

0 If the IDLE_A bit is set to zero, then the device server shall ignore the idle_a condition timer.

STANDBY_Z bit

1 If the STANDBY_Z bit is set to one, then the standby_z condition timer is enabled.

0 If the STANDBY_Z bit is set to zero, then the device server shall ignore the standby_z condition timer.

If any of the power condition enable bits (e.g., the IDLE_C bit or the STANDBY_Y bit) are set to zero and are not changeable (see 3.13.1.2), then the device server does not implement the power condition timer associated with that enable bit (see SPC-4).

IDLE_A CONDITION TIMER field

The IDLE_A CONDITION TIMER field specifies the initial value, in 100 millisecond increments, for the idle_a power condition timer (see SPC-4). This value may be rounded up or down to the nearest implemented time as described in 2.3.

STANDBY_Z CONDITION TIMER field

The STANDBY_Z CONDITION TIMER TIMER field specifies the initial value, in 100 millisecond increments, for the standby_z power condition timer (see SPC-4). This value may be rounded up or down to the nearest implemented time as described in 2.3.

IDLE_B CONDITION TIMER field

The IDLE_B CONDITION TIMER field specifies the initial value, in 100 millisecond increments, for the idle_b power condition timer (see SPC-4). This value may be rounded up or down to the nearest implemented time as described in 2.3.

IDLE_C CONDITION TIMER field

The IDLE_C CONDITION TIMER field specifies the initial value, in 100 millisecond increments, for the idle_c power condition timer (see SPC-4). This value may be rounded up or down to the nearest implemented time as described in 2.3.

STANDBY_Y CONDITION TIMER field

The STANDBY_Y CONDITION TIMER field specifies the initial value, in 100 millisecond increments, for the standby_y power condition timer (see SPC-4). This value may be rounded up or down to the nearest implemented time as described in 2.3.

4.3.15 Read-Write Error Recovery mode page (01h)

The Read-Write Error Recovery mode page (see table 287) specifies the error recovery parameters the device server shall use during any command that performs a read or write operation to the medium (e.g., READ commands, WRITE commands, and WRITE AND VERIFY commands).

Table 287. Read-Write Error Recovery mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	PAGE CODE (01h)					
1	PAGE LENGTH (0Ah)							
2	AWRE	ARRE	TB	RC	ERROR RECOVERY BITS			
					EER	PER	DTE	DCR
3	READ RETRY COUNT							
4	Obsolete							
5	Obsolete							
6	Obsolete							
7	Reserved						RESTRICTED FOR MMC-6	
8	WRITE RETRY COUNT							
9	Reserved							
10	(MSB)	RECOVERY TIME LIMIT						
11								(LSB)

PS (Parameters Savable) bit

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

AWRE (Automatic Write Reallocation Enabled) bit

0 An automatic write reallocation enabled (AWRE) bit set to zero specifies that the device server shall not perform automatic reallocation of defective logical blocks during write operations.

1 An AWRE bit set to one specifies that the device server shall enable automatic reallocation of defective logical blocks during write operations. The automatic reallocation shall be performed only if the device server has the valid data (e.g., original data in a buffer or recovered from the medium). The valid data shall be placed in the reallocated logical block. The device server shall report any failures that occur during the reallocation operation. Error reporting as specified by the error recovery bits (i.e., the EER bit, the PER bit, the DTE bit, and the DCR bit) shall be performed only after completion of the reallocation. See the REASSIGN BLOCKS command (see 3.31) for error procedures.

ARRE (Automatic Read Reallocation Enabled) bit

0 An automatic read reallocation enabled (ARRE) bit set to zero specifies that the device server shall not perform automatic reallocation of defective logical blocks during read operations.

1 An ARRE bit set to one specifies that the device server shall enable automatic reallocation of defective logical blocks during read operations. All error recovery actions required by the error recovery bits (i.e., the EER bit, the PER bit, the DTE bit, and the DCR bit) shall be processed. The automatic reallocation shall then be performed only if the device server successfully recovers the data. The recovered data shall be placed in the reallocated logical block. The device server shall report any failures that occur during the reallocation operation. Error reporting as specified by the error recovery bits (i.e., the EER bit, the PER bit, the DTE bit, and the DCR bit) shall be performed only after completion of the reallocation operation. See the REASSIGN BLOCKS command (see 3.31) for error procedures.

TB (Transfer Block) bit

0 A transfer block (TB) bit set to zero specifies that the device server shall not transfer a logical block to the data-in buffer if the logical block is not recovered within the recovery limits specified.

1 A TB bit set to one specifies that the device server shall transfer a logical block to the data-in buffer before returning CHECK CONDITION status if the logical block is not recovered within the recovery limits specified. The data returned in this case is vendor-specific. The TB bit does not affect the action taken for recovered data.

RC (Read Continuous) bit

0 A read continuous (RC) bit set to zero specifies that error recovery operations that cause delays are acceptable during the data transfer. Data shall not be fabricated.

1 An RC bit set to one specifies the device server shall transfer the entire requested length of data without adding delays to perform error recovery procedures. This implies that the device server may send data that is erroneous or fabricated in order to maintain a continuous flow of data. The device server shall assign priority to the RC bit over conflicting bits within this byte.

Note. Fabricated data may be data already in a buffer or any other vendor-specific data. The RC bit may be used in image processing, audio, or video applications.

EER (Enable Early Recovery) bit

1 An enable early recovery (EER) bit set to one specifies that the device server shall use the most expedient form of error recovery first.

0 An EER bit set to zero specifies that the device server shall use an error recovery procedure that minimizes the risk of error mis-detection or mis-correction. This bit only applies to data error recovery and it does not affect positioning retries.

Note. An EER bit set to one may imply an increase in the probability of error mis-detection or mis-correction. An EER bit set to zero allows the specified retry limit to be exhausted prior to using error correction codes.

PER (Post Error) bit

1 A post error (PER) bit set to one specifies that if a recovered read error occurs during a command performing a read or write operation, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to RECOVERED ERROR. If the DTE bit is set to one, then the PER bit shall be set to one

0 A PER bit set to zero specifies that if a recovered read error occurs during a command performing a read or write operation, then the device server shall perform error recovery procedures within the limits established by the error recovery parameters and only terminate the command with CHECK CONDITION status if the error becomes uncorrectable based on the established limits.

DTE (Data Terminate On Error) bit

1 A data terminate on error (DTE) bit set to one specifies that the device server shall terminate the data-in or data-out buffer transfer upon detection of a recovered error.

0 A DTE bit set to zero specifies that the device server shall not terminate the data-in or data-out buffer transfer upon detection of a recovered error.

DCR (Disable Correction) bit

1 A disable correction (DCR) bit set to one specifies that ECC shall not be used for data error recovery.

0 A DCR bit set to zero allows the use of ECC for data error recovery. If the EER bit is set to one, the DCR bit shall be set to zero.

The combinations of the error recovery bits (i.e., the EER bit, the PER bit, the DTE bit, and the DCR bit) are explained in table 288.

Table 288. Combined error recovery bit descriptions (Sheet 1 of 4)

EER	PER	DTE	DCR	Description
0	0	0	0	The device server shall perform the full number of retries as specified in the READ RETRY COUNT field for read operations, the WRITE RETRY COUNT field for write operations, and the VERIFY RETRY COUNT field (see 4.3.20) for verify operations and shall perform error correction in an attempt to recover the data. The device server shall not report recovered errors. The device server shall terminate a command with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurs during a read operation, the data in the block with the unrecoverable error may or may not be transferred to the data-in buffer depending on the setting of the transfer block (TB) bit.
0	0	0	1	The device server shall perform the full number of retries as specified in the READ RETRY COUNT field for read operations, the WRITE RETRY COUNT field for write operations, and the VERIFY RETRY COUNT field (see 4.3.20) for verify operations but shall not perform error correction in an attempt to recover the data. The device server shall not report recovered errors. The device server shall terminate a command with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurs during a read operation, the data in the block with the unrecoverable error may or may not be transferred to the data-in buffer depending on the setting of the transfer block (TB) bit.
0	0	1	0	Invalid mode. The PER bit shall be set to one if the DTE bit is set to one.
0	0	1	1	Invalid mode. The PER bit shall be set to one if the DTE bit is set to one.

Table 288. Combined error recovery bit descriptions (Sheet 2 of 4)

EER	PER	DTE	DCR	Description
0	**1**	**0**	**0**	The device server shall perform the full number of retries as specified in the READ RETRY COUNT field for read operations, the WRITE RETRY COUNT field for write operations, and the VERIFY RETRY COUNT field (see 4.3.20) for verify operations and shall perform error correction in an attempt to recover the data. The device server shall terminate a command with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecovered read error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer. If a recovered error occurs while the device server is performing a read or write operation, then, after the operation is complete, the device server shall terminate the command with CHECK CONDITION status with the sense key set to RECOVERED ERROR. The INFORMATIO field in the sense data shall contain the LBA of the last recovered error that occurred during the command.
0	**1**	**0**	**1**	The device server shall perform the full number of retries as specified in the READ RETRY COUNT field for read operations, the WRITE RETRY COUNT field for write operations, and the VERIFY RETRY COUNT field (see 4.3.20) for verify operations but shall not perform error correction in an attempt to recover the data. The device server shall terminate a command performing a read or write operation with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecovered read error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer. If a recovered error occurs while the device server is performing a read or write operation, then, after the operation is complete, the device server shall terminate the command with CHECK CONDITION status with the sense key set to RECOVERED ERROR. The INFORMATION field in the sense data shall contain the LBA of the last recovered error that occurred during the command.
0	**1**	**1**	**0**	The device server shall perform the full number of retries as specified in the READ RETRY COUNT field for read operations, the WRITE RETRY COUNT field for write operations, and the VERIFY RETRY COUNT field (see 4.3.20) for verify operations and shall perform error correction in an attempt to recover the data. The device server shall terminate a command performing a read or write operation with CHECK CONDITION status before the transfer count is exhausted if any error, either recoverable or unrecoverable, is detected. The INFORMATION field in the sense data shall contain the LBA of the block in error. If an unrecoverable data error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer.

Table 288. Combined error recovery bit descriptions (Sheet 3 of 4)

EER	PER	DTE	DCR	Description
0	1	1	1	The device server shall perform the full number of retries as specified in the READ RETRY COUNT field for read operations, the WRITE RETRY COUNT field for write operations, and the VERIFY RETRY COUNT field (see 4.3.20) for verify operations but shall not perform error correction in an attempt to recover the data. The device server shall terminate a command performing a read or write operation with CHECK CONDITION status before the transfer count is exhausted if any error, either recoverable or unrecoverable, is detected. The INFORMATION field in the sense data shall contain the LBA of the block in error. If an unrecoverable data error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer.
1	0	0	0	The device server shall perform the fewest possible number of retries and perform error correction in an attempt to recover the data. The device server shall not report recovered errors. The device server shall terminate a command performing a read or write operation with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer.
1	0	0	1	Invalid mode. The DCR bit shall be set to zero if the EER bit is set to one.
1	1	0	0	The device server shall perform the fewest possible number of retries and perform error correction in an attempt to recover the data. The device server shall terminate a command performing a read or write operation with CHECK CONDITION status before the transfer count is exhausted only if an unrecoverable error is detected. If an unrecoverable data error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer If a recovered error occurs while the device server is performing a read or write operation, then, after the operation is complete, the device server shall terminate the command with CHECK CONDITION status with the sense key set to RECOVERED ERROR. The INFORMATION field in the sense data shall contain the LBA of the last recovered error that occurred during the command.
1	1	0	1	Invalid mode. The DCR bit shall be set to zero if the EER bit is set to one.

Table 288. Combined error recovery bit descriptions (Sheet 4 of 4)

EER	PER	DTE	DCR	Description
1	**1**	**1**	**0**	The device server shall perform the fewest possible number of retries and perform error correction in an attempt to recover the data. The device server shall terminate the command performing a read or write operation with CHECK CONDITION status before the transfer count is exhausted if any error, either recoverable or unrecoverable, is detected. The INFORMATION field in the sense data shall contain the LBA of the block in error. If an unrecoverable data error occurs during a read operation, the transfer block (TB) bit determines whether the data for the logical block with the unrecovered read error is transferred to the data-in buffer.
1	**1**	**1**	**1**	Invalid mode. The DCR bit shall be set to zero if the EER bit is set to one.
If an invalid combination of the error recovery bits is sent by the application client the device server shall terminate the MODE SELECT 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.				

READ RETRY COUNT field

The READ RETRY COUNT field specifies the number of times that the device server shall attempt its recovery algorithm during read operations.

WRITE RETRY COUNT field

The WRITE RETRY COUNT field specifies the number of times that the device server shall attempt its recovery algorithm during write operations.

RECOVERY TIME LIMIT field

The RECOVERY TIME LIMIT field specifies in milliseconds the maximum time duration that the device server shall use for data error recovery procedures. The device server may round this value as described in 2.3. The limit in this field specifies the maximum error recovery time allowed for any individual command. A RECOVERY TIME LIMIT field set to zero specifies that the device server shall use its default value.

When both a retry count and a recovery time limit are specified, the field that specifies the recovery action of least duration shall have priority.

To disable all types of correction and retries the application client should set the EER bit to zero, the PER bit to one, the DTE bit to one, the DCR bit to one, the READ RETRY COUNT field to 00h, the WRITE RETRY COUNT field to 00h, and the RECOVERY TIME LIMIT field to 0000h.

4.3.16 Protocol Specific Logical Unit mode page (18h)

4.3.16.1 Protocol specific parameters

4.3.16.1.1 Protocol specific parameters introduction

Some commands use protocol specific information in their CDBs or parameter lists. This subclause describes those protocol specific parameters.

The Protocol Specific Logical Unit mode page (see table 289) provides protocol specific controls that are associated with a logical unit.

For Parallel SCSI, see 4.3.16.

For Fibre Channel (FC), see 4.3.16.2.

For Serial Attached SCSI (SAS), see 4.3.16.3.

Table 289. Protocol Specific Logical Unit mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (18h)					
1	PAGE LENGTH (n-1)							
2	Reserved				PROTOCOL IDENTIFIER			
3	PROTOCOL SPECIFIC MODE PARAMETERS							
n								

During an I_T_L nexus, the Protocol Specific Logical Unit mode page controls parameters that affect both:

a) One or more target ports; and

b) The logical unit.

The parameters that may be implemented are specified in the SCSI transport protocol standard for the target port. The mode page policy (see 4.4.10) for this mode page shall be shared or per target port and should be per target port.

The parameters for a target port and logical unit affect their behavior regardless of which initiator port is forming an I_T_L nexus with the target port and logical unit. If a parameter value is changed, the device server shall establish a unit attention condition for the initiator port associated with every I_T nexus except the I_T nexus on which the MODE SELECT command was received, with the additional sense code set to MODE PARAMETERS CHANGED.

The PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

PROTOCOL IDENTIFIER field

The value in the PROTOCOL IDENTIFIER (see table 290) field defines the SCSI transport protocol to which the mode page applies. For a MODE SENSE command (see 3.13), the device server shall set the PROTOCOL IDENTIFIER field to one of the values shown in Table 290 to indicate the SCSI transport protocol used by the target port through which the MODE SENSE command is being processed. For a MODE SELECT command (see 3.11), the application client shall set the PROTOCOL IDENTIFIER field to one of the values shown in Table 290 indicating the SCSI transport protocol to which the protocol specific mode parameters apply. If a device server receives a mode page containing a transport protocol identifier value other than the one used by the target port on which the MODE SELECT command was received, then the 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 PARAMETER LIST.

Protocol specific parameters may include a PROTOCOL IDENTIFIER field (see table 290) as a reference for the SCSI transport protocol to which the protocol specific parameter applies.

Table 290. PROTOCOL IDENTIFIER values

Protocol Identifier	Description	Protocol Standard
0h	Fibre Channel	FCP
1h	Parallel SCSI	SPI-5
2h	SSA	SSA-S3P
3h	IEEE 1394	SBP-3
4h	SCSI Remote Direct Memory Access Protocol	SRP
5h	Internet SCSI (iSCSI)	iSCSI
6h	SAS Serial SCSI Protocol	SAS
7h	Automation/Drive Interface Transport Protocol	ADT
8h	AT Attachment Interface (ATA/ATAPI)	ATA/ATAPI-7
9h - Eh	Reserved	
Fh	No specific protocol	

4.3.16.2 Fibre Channel Logical Unit Control mode page (18h)

The Fibre Channel Logical Unit Control mode page (see table 291) contains those parameters that select FCP logical unit operation options. The implementation of any parameter and its associated functions is optional. The mode page follows the MODE SENSE and MODE SELECT command rules.

Table 291. Fibre Channel Logical Unit Control mode page (18h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0B)	PAGE CODE (18h)					
1	PAGE LENGTH (06h)							
2	RESERVED				PROTOCOL IDENTIFIER (FCP = 0h)			
3	Reserved							EPDC
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							

EPDC (ENABLE PRECISE DELIVERY CHECKING) bit

1 An ENABLE PRECISE DELIVERY CHECKING (EPDC) bit of one indicates that the logical unit shall use the precise delivery function (see FCP-4).

0 When the EPDC bit is set to zero, the logical unit shall not use the precise delivery function and shall ignore the contents of the CRN field in the FCP_CMND IU.

The EPDC bit is valid for all types of link connections. If the precise delivery function is not supported and the Fibre Channel Logical Unit Control mode page is supported by the logical unit, the EPDC bit shall be masked as not changeable and shall follow the MODE SENSE and MODE SELECT command rules (see 3.11).

If the Fibre Channel Logical Unit Control mode page is not supported by a logical unit, the initiator shall assume that the precise delivery function is not supported by that logical unit.

4.3.16.3 SAS Protocol-Specific Logical Unit mode page

The SAS Protocol-Specific Logical Unit mode page (see SPC-4) contains parameters that affect SSP target port operation on behalf of the logical unit.

Table 292 defines the subpages of this mode page.

Table 292. SAS Protocol-Specific Logical Unit mode page subpages

Subpage	Description	Reference
Short page	Short format	4.3.16.3.1
Long page 00h	Not allowed	
Long page E0h - FEh	Vendor specific	
Long page FFh	Return all subpages for the Protocol-Specific Logical Unit mode page	SPC-4
All others	Reserved	

4.3.16.3.1 Protocol-Specific Logical Unit mode page - short format (18h)

The mode page policy (see 4.4.10) for the Protocol-Specific Logical Unit mode page short format subpage shall be either shared or per target port. If a SAS target device has multiple SSP target ports, the mode page policy should be per target port. Parameters in this page shall affect all phys in the SSP target port if the mode page policy is per target port, and shall affect all SSP target ports in the SAS target device if the mode page policy is shared.

Table 293 defines the format of the page for SAS SSP.

Table 293. Protocol-Specific Logical Unit mode page for SAS SSP - short format

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (18h)					
1	PAGE LENGTH (06h)							
2	Reserved			TRANS-PORT LAYER RETRIES	PROTOCOL IDENTIFIER (6h)			
3	Reserved							
4 - 7	Reserved							

The PARAMETERS SAVEABLE (PS) bit is defined in 4.3.5.

The SPF bit shall be set to zero for access to the short format mode page.

The PAGE CODE field shall be set to 18h.

The PAGE LENGTH field shall be set to 06h.

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

TRANSPORT LAYER RETRIES bit

1 A TRANSPORT LAYER RETRIES bit set to one specifies that the target port shall support transport layer retries for XFER_RDY and DATA frames for the logical unit as described in SAS-2 Transport Layer.

0 A TRANSPORT LAYER RETRIES bit set to zero specifies that transport layer retries shall not be used.

4.3.17 Protocol Specific Port mode page (19h)

The Protocol Specific Port mode page provides protocol specific controls that are associated with a SCSI port. The page_0 format (see table 294) is used for subpage 00h and SUB_PAGE format (see table 295) is used for subpages 01h through FEh. See the SCSI transport protocol standard for definition of the protocol specific mode parameters.

Table 294. Page_0 format Protocol Specific Port mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (n-1)							
2	Reserved				PROTOCOL IDENTIFIER			
3	PROTOCOL SPECIFIC MODE PARAMETERS							
n								

Table 295. Sub_page format Protocol Specific Port mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (19h)					
1	SUBPAGE CODE							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER			
6	PROTOCOL SPECIFIC MODE PARAMETERS							
n								

The Protocol Specific Port mode page controls parameters that affect one or more target ports. The parameters that may be implemented are specified in the SCSI transport protocol standard for the target port. The mode page policy (see 4.4.10) for this mode page shall be shared or per target port. If the SCSI target device contains more than one target port, the mode page policy should be per target port.

The parameters for a target port affect its behavior regardless of which initiator port is forming an I_T nexus with the target port. The parameters may be accessed by MODE SENSE (see 3.13) and MODE SELECT (see 3.11) commands directed to any logical unit accessible through the target port. If a parameter value is changed, the device server for all logical units accessible through the target port shall establish a unit attention condition for the initiator port associated with every I_T nexus except the I_T nexus on which the MODE SELECT command was received, with the additional sense code set to MODE PARAMETERS CHANGED.

The PS bit, SPF bit, PAGE CODE field, SUBPAGE CODE field, and PAGE LENGTH field are described in 4.3.5.

The value in the PROTOCOL IDENTIFIER field (see 4.3.16.1.1) defines the SCSI transport protocol to which the mode page applies. For a MODE SENSE command, the device server shall set the PROTOCOL IDENTIFIER field to one of the values shown in table 290 to indicate the SCSI transport protocol used by the target port through which the MODE SENSE command is being processed. For a MODE SELECT command, the application client shall set the PROTOCOL IDENTIFIER field to one of the values shown in table 290 indicating the SCSI transport protocol to which the protocol specific mode parameters apply. If a device server receives a mode page containing a transport protocol identifier value other than the one used by the target port on which the MODE SELECT command was received, then 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 PARAMETER LIST.

For Parallel SCSI, see 4.3.17.1.

For Fibre Channel (FC), see 4.3.17.2.

For Serial Attached SCSI (SAS), see 4.3.17.3.

4.3.17.1 Port Control Mode page (19h) for Parallel SCSI

The Port Control Mode page contains the parameters that affect SCSI target port operation options. The page shall be implemented by LUN 0 of all SPI SCSI target devices. The page shall not be implemented by logical units other than LUN 0. The implementation of any bit and its associated functions is optional. The page follows the MODE SENSE/MODE SELECT rules specified by the ANSI SPC-4 standard.

Each SCSI target port shall maintain an independent set of port control mode page parameters for each SCSI initiator port. The parameters saveable bit in the mode page format header returned with MODE SENSE command shall be set to zero if the long mode page format is being used (i.e., Long bit set to one), indicating the parameters are not saved through resets.

After a MODE SELECT command, parameter settings shall remain in effect until either:

(a) settings are changed by another MODE SELECT command,

(b) a logical unit reset of LUN 0 occurs,

(c) an SDTR negotiation successfully completes,

(d) a WDTR negotiation successfully completes, or

(e) a PPR negotiation successfully completes with the HOLD_MCS bit set to zero.

Table 296. Port Control Mode page short format (19h)

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Long (0)	PAGE CODE (19h)					
1	PAGE LENGTH (06h)							
2	Reserved				PROTOCOL IDENTIFIER (1h)			
3	Reserved							
4	(MSB)	SYNCHRONOUS TRANSFER TIMEOUT						
5								(LSB)
6	Reserved							
7	Reserved							

PS (Parameter Savable) bit

The Parameter Savable (PS) bit of 0 indicates that the page 19h parameter data is not savable in non-volatile memory.

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field value of 1h indicates that this mode page applies to a SPI SCSI device. See the SPC-4 standard for other port control page protocol identifiers.

SYNCHRONOUS TRANSFER TIMEOUT field

The SYNCHRONOUS TRANSFER TIMEOUT field indicates the maximum amount of time in 1 ms increments that the SCSI target port shall wait before generating an error by doing an unexpected bus free. The SCSI target port shall only go to a BUS FREE phase if one of the following events causes the timer, once started, to not reset or reload before expiring.

a) If there is a REQ transition when there are no outstanding REQs waiting for an ACK then load and start the timer.

b) If there is a REQ transition when there are any outstanding REQs waiting for an ACK then there is no effect on the timer.

c) If there is an ACK transition when there are outstanding REQs waiting for an ACK then load and start the timer.

d) If, after an ACT transition, there are no outstanding REQs waiting for an ACK then stop the timer.

A SYNCHRONOUS TRANSFER TIMEOUT field value of 0000h indicates that the function is disabled.

A value of FFFFh indicates an unlimited period.

Table 297. Port Control Mode page long format (19h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	Long (1)	PAGE CODE (19h)					
1	SUBPAGE CODE							
2	(MSB)	PAGE LENGTH (n – 3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER = 1h			
6	PROTOCOL-SPECIFIC MODE PARAMETERS							
n								

The Parameter Savable (PS) bit of 0 indicates that the page 19h parameter data is not savable in non-volatile memory.

The SUBPAGE CODE field indicates which subpage is being accessed. If the SUBPAGE CODE field is zero, the SCSI target device shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code set to ILLEGAL FIELD IN PARAMETER LIST.

Subpage Code	Description	Reference clause
1	Margin control subpage	4.3.17.1.1
2	Saved training configuration values subpage	4.3.17.1.2
3	Negotiated settings subpage	4.3.17.1.3
4	Report transfer capabilities subpage	4.3.17.1.4

The PROTOCOL IDENTIFIER field value of 1h indicates that this mode page applies to a SPI SCSI device. See the SPC-4 standard for other port control page protocol identifiers.

4.3.17.1.1 Margin control subpage

The margin control subpage contains parameters that set and report margin control values for usage between the SCSI initiator/target port pair on subsequent synchronous and paced transfers.

A MODE SENSE command shall return the current settings for the SCSI initiator/target port pair. Fields that are not implemented shall be reported as zero.

Note. The default value of each margin control field should be 0000b.

The margin control fields indicate absolute conditions centered around their default values. Absolute conditions means that the previous history of the parameter has no relevance to the value of the parameter.

The maximum supported setting for each field is 0111b and the minimum supported setting for each field is 1000b. Up to 16 distinct values are available for each field, representing monotonically changing device response. Devices that support fewer than 16 distinct values for a field should round non-supported settings to a supported value.

The actual response of a SCSI device to a field value is vendor-specific and calibration of the actual minimum and maximum responses to different field values is not defined in this manual or the ANSI SPI-4 specification. Margin control settings should not cause the driver to violate SPI-4 specification electrical limits. Margin control settings should affect only the REQUEST, ACKNOWLEDGE, DATA BUS, P_CRCA, and DB(P1) signals and should affect all of these signals driven by the device by the same amount.

The margin control fields contain two's complement values as shown in the table below.

Value		
Binary	**Decimal**	**Parameter values**
0111b	7	Maximum setting
0110b	6	
0101b	5	
0100b	4	
0011b	3	
0010b	2	
0001b	1	
000b	0	Recommended default value
111b	–1	
1110b	–2	
1101b	–3	
1100b	–4	
1011b	–5	
1010b	–6	
1001b	–7	
1000b	–8	Minimum setting

Table 298. Margin control subpage

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	DRIVER STRENGTH				Reserved			
2	DRIVER ASYMMETRY				DRIVER PRECOMPENSATION			
3	DRIVER SLEW RATE				Reserved			
4	Reserved							
5	Reserved							
6	Reserved							
7	VENDOR-SPECIFIC							
8 : 15	Reserved							

DRIVER STRENGTH field

The DRIVER STRENGTH field indicates the relative amount of driver source current used by the driver (see clause 7.3.2 in the ANSI SPI-4 specification). The DRIVER STRENGTH field affect both the strong and weak drivers. A larger value indicates more driver source current.

DRIVER ASYMMETRY field

The DRIVER ASYMMETRY field indicates the relative difference between the amplitudes of asserted and negated signals launched from the driver. A larger value indicates a relatively stronger asserted signal compared to the negated signal.

DRIVER PRECOMPENSATION field

The DRIVER PRECOMPENSATION field indicates the relative difference between the weak driver and the strong driver amplitudes when pre-compensation is enabled. A larger value indicates a larger difference between the weak and strong amplitudes.

DRIVER SLEW RATE field

The DRIVER SLEW RATE field indicates the relative difference between the assertion and negation magnitudes divided by the rise or fall time. A larger value indicates a faster slew rate.

4.3.17.1.2 Saved training configuration values subpage

The saved training configuration values subpage is used to report the SCSI device's saved training configuration values. These vendor-specific values are maintained by the SCSI device when the retain training information option is enabled. The fields are listed in Table 299 however the content of the field is vendor-specific.

Only values for the current I_T nexus are reported.

Table 299. Saved training configuration values subpage

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
3								
4	MSB	DB(0) Value						
7								LSB
.	.							
.	.							
64	MSB	DB(15) Value						
67								LSB
68	MSB	P_CRCA Value						
71								LSB
72	MSB	P1 Value						
75								LSB
76	MSB	BSY Value						
79								LSB
80	MSB	SEL Value						
83								LSB
84	MSB	RST Value						
87								LSB
88	MSB	REQ Value						
91								LSB
92	MSB	ACK Value						
95								LSB

Bit Byte	7	6	5	4	3	2	1	0
96	MSB							
				ATN Value				
99								LSB
100	MSB							
				C/D Value				
103								LSB
104	MSB							
				I/O Value				
107								LSB
108	MSB							
				MSG Value				
111								LSB
112								
				Reserved				
127								

4.3.17.1.3 Negotiated settings subpage

The negotiated setting subpage is used to report the negotiated settings of a SCSI target port for the current I_T nexus.

Table 300. Negotiated settings subpage

Bit Byte	7	6	5	4	3	2	1	0
0	TRANSFER PERIOD FACTOR							
1	Reserved							
2	REQ/ACK OFFSET							
3	TRANSFER WIDTH EXPONENT							
4	Reserved	PROTOCOL OPTIONS BITS						
5	Reserved				TRANSCEIVER MODE		SENT PCOMP_EN	RECEIVED PCOM_EN
6	Reserved							
7	Reserved							

TRANSFER PERIOD FACTOR field

The TRANSFER PERIOD FACTOR field indicates the negotiated transfer period factor for the current I_T nexus.

REQ/ACK OFFSET field

The REQ/ACK OFFSET field indicates the negotiated REQ/ACK offset for the current I_T nexus.

TRANSFER WIDTH EXPONENT field

The TRANSFER WIDTH EXPONENT field indicates the negotiated transfer width exponent for the current I_T nexus.

PROTOCOL OPTIONS BITS field

The PROTOCOL OPTIONS BITS field contains the negotiated protocol options for the current I_T nexus.

RECEIVED PCOMP_EN bit

The RECEIVED PCOMP_EN bit contains the value of the PCOMP_EN bit received by the SCSI target port for the current I_T nexus.

SENT PCOMP_EN bit

The SENT PCOMP_EN bit contains the value of the PCOMP_EN bit sent by the SCSI target port for the current I_T nexus.

TRANSCEIVER MODE field

The TRANSCEIVER MODE field specifies the current bus mode of the SCSI target port as defined below.

Code	Bus mode
00b	Unknown (e.g., device not capable of reporting bus mode)
01b	Single-ended
10b	Low Voltage Differential
11b	High Voltage Differential

4.3.17.1.4 Report transfer capabilities subpage

The Report transfer capabilities subpage is used to report the transfer capabilities for the SCSI target port. The values in this subpage are not changeable via a MODE SELECT command.

Table 301. Report transfer capabilities subpage

Bit Byte	7	6	5	4	3	2	1	0
0	MINIMUM TRANSFER PERIOD FACTOR							
1	Reserved							
2	MAXIMUM REQ/ACK OFFSET							
3	MAXIMUM TRANSFER WIDTH EXPONENT							
4	PROTOCOL OPTIONS BITS SUPPORTED							
5	Reserved							
6	Reserved							
7	Reserved							

MINIMUM TRANSFER PERIOD FACTOR byte

The MINIMUM TRANSFER PERIOD FACTOR field shall be set to the smallest value of the transfer period factor supported by the SCSI target port.

MAXIMUM REQ/ACK OFFSET byte

The MAXIMUM REQ/ACK OFFSET shall be set to the largest value of the REQ/ACK offset supported by the SCSI target port.

MAXIMUM TRANSFER WIDTH EXPONENT byte

The MAXIMUM TRANSFER WIDTH EXPONENT field shall be set to the largest value of the transfer width exponent supported by the SCSI target port.

PROTOCOL OPTIONS BITS SUPPORTED byte

The SCSI target port shall set the bits in the PROTOCOL OPTIONS BITS SUPPORTED field to indicate the protocol options supported by the SCSI target port.

4.3.17.2 Port Control Mode page (19h) for Fibre Channel (FC)

The Fibre Channel Interface Control page controls options relevant to Fibre Channel protocol. It is intended for the control of features unique to Fibre Channel protocol that are not suitable for control by login or other techniques defined for Fibre Channel.

Both the current and saved values of Mode page 19h, byte 3, are changeable using the Mode Select command. When the saved value of Byte 3 is changed, a new value is stored in nonvolatile (flash) memory, not on disc, and is reported during a Mode Sense command when either the current or saved value is requested.

The current value of Byte 3 may be modified using Mode Select command any number of times. The saved value of Byte 3 may be modified using the Mode Select command up to 32 times. After 32 modifications, additional attempts to modify will result in a response of Illegal Request (05) with Error Code of 2600 (Invalid field in parameter list) and the Sense Key pointing to byte 3 as the offending parameter. Additional updates of Byte 3 saved value may be accomplished by downloading new microcode that supports Fibre Channel Interface Control page 19h. After a download, the saved value of Byte 3 may be updated 31 more times or 32 times if the value is 0 at the time of download.

The saved and current value of Byte 3 is preserved through the download.

Table 302. Fibre Channel Interface Control page (19h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (06h)							
2	Reserved				PROTOCOL IDENTIFIER (FCP = 0h)			
3	DTFD	PLPB	DDIS	DLM	RHA	ALWLI	DTIPE	DTOLI
4	Reserved							
5	Reserved							
6	Reserved							
7	SEQUENCE INITIATIVE RESOURCE RECOVERY TIMEOUT VALUE (RR_TOVSEQ_INIT)							

PS (Parameter Savable)

The PS bit is defined in 4.3.5.

SPF bit

The SPF field shall be set to zero for access to the short format mode page.

Page Length byte

The length of the Fibre Channel Interface Control page in bytes (06h). If the allocation length is too small to transfer all of the page, the page length is not adjusted to reflect the truncation.

DTFD (Disable Target Fabric Discovery)

1 The target attached by an FC-AL loop will not recognize the presence of a fabric loop port (FL_Port) on the loop. The target will only perform the private loop functions defined for FC-PLDA targets.

0 The target attached by an FC-AL loop will discover the FL_Port if present on the loop and will perform the public loop functions defined for FC-FLA targets. Targets attached to an N_Port or to an F_Port will ignore this bit.

PLPB (Prevent Loop Port Bypass)

1 The target ignores any Loop Port Bypass (LPB) and Loop Port Enable (LPE) primitive sequences. The loop port remains enabled.

Note. Do not set the PLPB bit to one (1) and the DTIPE bit to one (1) at the same time as this is an illegal bit combination. When an illegal bit combination is sent by the application client, the device server returns Check Condition status and sets the sense key to Illegal Request with the additional sense code set to Invalid Field in the Parameter List.

0 The target allows the Loop Port Bypass and Port Bypass Enable primitive sequences to control the port bypass circuit.

DDIS (Disable Discovery)

1 The target does not require receipt of Address or Port Discovery ELS following loop initialization. The target resumes processing of tasks upon completion of loop initialization.

The target must wait to receive an Address or Port Discovery ELS before it resumes processing tasks for that initiator.

DLM (Disable Loop Master)

1 Indicates that a target attached to an FC-AL-2 loop shall not participate in loop master arbitration and shall not become loop master. The target shall only repeat LISM frames it receives.

0 The target may participate in loop master arbitration in the normal manner and, if successful, may become loop master during the loop initialization process. Targets not attached to an arbitrated loop shall ignore the DLM bit.

REQUIRE HARD ADDRESS (RHA) bit - formerly DSA bit

If the REQUIRE HARD ADDRESS (RHA) bit is set to one, a target FCP_Port attached to an arbitrated loop (see FCAL-2) shall only attempt to obtain its hard address available in the SCA-2 connector (see SFF-8067) or device address jumpers during loop initialization. The target FCP_Port shall not attempt to obtain an address during the LISA phase of initialization (see FC-AL-2). If there is a conflict for the hard address selection during loop initialization or the target FCP_Port does not have a valid hard address available, the target FCP_Port shall enter the nonparticipating state. If the target FCP_Port detects loop initialization while in the nonparticipating state, the target FCP_Port shall again attempt to get its hard address. If the hard address has not changed from the address obtained in a previous successful loop initialization, the target FCP_Port shall attempt to obtain the address in the LIFA phase if a valid Fabric Login exists or LIPA phase of loop initialization. If the hard address has changed, the target FCP_Port shall attempt to obtain the new address in the LIHA phase.

If the RHA bit is set to zero, the target FCP_Port follows the normal initialization procedure, including the possibility of obtaining a soft address during the loop initialization process.

Target FCP_Ports not attached to an arbitrated loop shall ignore the RHA bit.

DSA (Disable Soft Address)

1 The target does not select a soft address if there is a conflict for the Select_ID address available in the SCA connector during loop initialization. If there is a Select_ID address conflict, the target enters the non-participation state. If the target detects loop initialization while in the non-participation state, the target again attempts to get the Select_ID address.

0 The target attempts to obtain a soft address during loop initialization if the Select_ID address is not available or if the Select_ID address indicates the use of a soft address (7Fh).

If ESI activity is underway when the request for the hard address is received, the drive shall use the last known value of the hard address before the current ESI activity started.

ALWLI (Allow Login Without Loop Initialization)

1 The target uses the Select_ID address available in the SCA connector and accepts logins without verifying the address with loop initialization.

0 The target must verify the address through loop initialization before accepting a login.

DTIPE (Disable Target Initiated Port Enable)

1 The target waits for an initiator to send the Loop Port Enable primitive before inserting itself into the loop. The target uses the Select_ID address available in the SCA connector to determine if primitives are addressed to it.

Note. Do not set the PLPB bit to one (1) and the DTIPE bit to one (1) at the same time as this is an illegal bit combination. When an illegal bit combination is sent by the application client, the device server returns Check Condition status and sets the sense key to Illegal Request with the additional sense code set to Invalid Field in the Parameter List.

0 After completing self test, the target enables the port in the loop without waiting for a Loop Port Enable primitive.

DTOLI (Disable Target Originated Loop Initialization)

1 The target does not originate the initializing LIP following insertion into the loop. The target responds to an Initializing LIP when it is received. The target originates the Loop Failure LIP if it detects loop failure at its input. The target originates the Initializing LIP when the loop failure is corrected.

0 After completing self test, the target originates the Initializing LIP when it enables a port in a loop.

SEQUENCE INITIATIVE RESOURCE RECOVERY TIMEOUT VALUE (RR_TOV$_{SEQ_INIT}$) field

The RR_TOV$_{SEQ_INIT}$ timer (see FCP-4) is defined by the RR_TOVSEQ_INIT field and the RR_TOV UNITS field.

The RR_TOV UNITS field indicates the units for the RR_TOV$_{SEQ_INIT}$ field value, according to table 303.

Table 303. Values for RR_TOV UNITS

Byte 6			Units of measure for RR_TOV$_{SEQ_INIT}$
bit 2	bit 1	bit 0	
0	0	0	No timer is specified
0	0	1	0.001 s
0	1	1	0.1 s
1	0	1	10 s
All other values			Reserved

The RR_TOV$_{SEQ_INIT}$ field indicates the number of time units specified by the RR_TOV UNITS field that shall be used by the timer that performs the RR_TOV$_{SEQ_INIT}$ timeout function. If no timer is specified, the RR_TOV$_{SEQ_INIT}$ value shall be ignored by the device server and a vendor specific default value shall be used for RR_TOV$_{SEQ_INIT}$.

4.3.17.3 Protocol Specific Port Mode page (19h) for Serial Attached SCSI (SAS)

4.3.17.3.1 Protocol-specific Port mode page overview

The Protocol-Specific mode page (see 4.3.17) contains parameters that affect SSP target port operation. If the mode page is implemented, all logical units in SCSI target devices in SAS domains supporting the MODE SELECT or MODE SENSE commands shall implement the page.

If a SAS target device has multiple SSP target ports, changes in the short page parameters for one SSP target port should not affect other SSP target ports.

Table 304 defines the subpages of this mode page.

Table 304. Protocol-Specific Port mode page subpages

Subpage	Description	Reference
Short page	Short format	4.3.17.3.2 and 4.3.17.3.3
Long page 00h	Not allowed	
Long page 01h	Phy Control and Discover subpage	4.3.17.3.4
Long page 02h	Port Control subpage	4.3.17.3.5
Long page 03h	Enhanced Phy Control subpage	4.3.17.3.6
Long page E5h	Transceiver Control (Out) subpage	4.3.17.3.7
Long page E6h	Transceiver Control (In) subpage	4.3.17.3.8
Long page FFh	Return all subpages for the Protocol-Specific Port mode page	SPC-4
All others	Reserved	

4.3.17.3.2 Protocol-Specific Port mode page - short format - SAS / SAS-1.1

This clause describes the Protocol-Specific Port mode page implemented by products that are SAS and SAS-1.1 compliant. The mode page policy (see 4.4.10) for the Protocol-Specific mode page short format subpage shall be either shared or per target port. If a SAS target device has multiple SSP target ports, the mode page policy should be per target port.

Parameters in this page shall affect all phys in the SSP target port if the mode page policy is per target port, and shall affect all SSP target ports in the SAS target device if the mode page policy is shared.

Table x defines the format of the page for SAS SSP.

Table 305. Protocol-Specific Port mode page for SAS SSP - short format - SAS / SAS-1.1

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (06h)							
2	Reserved			READY LED MEAN-ING	PROTOCOL IDENTIFIER (6h)			
3	Reserved							
4	(MSB)	I_T NEXUS LOSS TIME						
5								(LSB)
6	(MSB)	INITIATOR RESPONSE TIMEOUT						
7								(LSB)

The PARAMETERS SAVEABLE (PS) bit is defined in 4.3.5.

The SPF field shall be set to zero for access to the short format mode page.

The PAGE CODE field shall be set to 19h.

The PAGE LENGTH field shall be set to 06h.

READY LED MEANING bit

The READY LED MEANING bit specifies the READY LED signal behavior (see SAS-2). Regardless of the mode page policy (see 4.4.10) for this mode page, the shared mode page policy shall be applied to the READY LED MEANING bit.

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

I_T NEXUS LOSS TIME field

The I_T NEXUS LOSS TIME field contains the time that the SSP target port shall retry connection requests to an SSP initiator port that are rejected with responses indicating the SSP initiator port may no longer be present before recognizing an I_T nexus loss. table 306 defines the values of the I_T NEXUS LOSS TIME field. If this mode page is not implemented, the I_T nexus loss time is vendor specific. This value is enforced by the port layer.

Table 306. I_T NEXUS LOSS TIME values

Code	Description
0000h	Vendor-specific amount of time.
0001h to FFFEh	Time in milliseconds.
FFFFh	The SSP target port shall never recognize an I_T nexus loss (i.e., it shall retry the connection requests forever).

Note. If this mode page is implemented, the default value of the I_T NEXUS LOSS TIME field should be non-zero. It is recommended that this value be 2 000 ms.

INITIATOR RESPONSE TIMEOUT field

The INITIATOR RESPONSE TIMEOUT field contains the time in milliseconds that the SSP target port shall wait for the receipt of a frame (e.g., a write DATA frame) before aborting the command associated with that frame. An INITIATOR RESPONSE TIMEOUT field value of zero indicates that the SSP target port shall disable the initiator response timeout timer. If this mode page is not implemented, the logical unit shall not implement an initiator response timeout timer. This value is enforced by the transport layer.

4.3.17.3.3 Protocol-Specific Port mode page- short format - SAS-2

This clause describes the Protocol-Specific Port mode page implemented by products that are SAS-2 compliant. The Protocol-Specific Port mode page contains parameters that affect SSP target port operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT or MODE SENSE commands.

The mode page policy (see 4.4.10) for this mode page shall be either shared or per target port. If a SAS target device has multiple SSP target ports, then the mode page policy should be per target port.

Parameters in this mode page:

a) shall affect all phys in the SSP target port if the mode page policy is per target port; or
b) shall affect all SSP target ports in the SAS target device if the mode page policy is shared.

Table 307 defines the format of the page for SAS SSP.

Table 307. Protocol-Specific Port mode page for SAS SSP- short format - SAS-2

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (19h)					
1	PAGE LENGTH (0Eh)							
2	Reserved	CONTINUE AWT	BROADCAST ASYNCHRONOUS EVENT	READY LED MEANING	PROTOCOL IDENTIFIER (6h)			
3	Reserved							
4	(MSB)	I_T NEXUS LOSS TIME						
5								(LSB)
6	(MSB)	INITIATOR RESPONSE TIMEOUT						
7								(LSB)
8	(MSB)	REJECT TO OPEN LIMIT						
9								(LSB)
10	Reserved							
15								

The PARAMETERS SAVEABLE (PS) bit is defined in 4.3.5.

The SPF field shall be set to zero for access to the short format mode page.

The PAGE CODE field shall be set to 19h.

The PAGE LENGTH field shall be set to 0Eh.

CONTINUE AWT BIT

0 A CONTINUE AWT bit set to zero specifies that the SAS port shall stop the Arbitration Wait Time timer and set the Arbitration Wait Time timer to zero when it receives an OPEN_REJECT (RETRY).

1 A CONTINUE AWT bit set to one specifies that the SAS port shall not stop the Arbitration Wait Time timer and shall not set the Arbitration Wait Time timer to zero when the SAS port receives an OPEN_REJECT (RETRY).

BROADCAST ASYNCHRONOUS EVENT bit

0 A BROADCAST ASYNCHRONOUS EVENT bit set to zero specifies that the device server shall disable origination of Broadcast (Asynchronous Event).

1 A BROADCAST ASYNCHRONOUS EVENT bit set to one specifies that the device server shall enable origination of Broadcast (Asynchronous Event) (see SAS-2).

READY LED MEANING bit

The READY LED MEANING bit specifies the READY LED signal behavior (see SAS-2). Regardless of the mode page policy (see 4.4.10) for this mode page, the shared mode page policy shall be applied to the READY LED MEANING bit.

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field is defined in 4.3.16 and shall be set to the value defined in table 290 indicating that this is a SAS SSP specific mode page.

I_T NEXUS LOSS TIME field

The I_T NEXUS LOSS TIME field is defined in 4.3.17.3.2.I

INITIATOR RESPONSE TIMEOUT field

The INITIATOR RESPONSE TIMEOUT field is defined in 4.3.17.3.2.

REJECT TO OPEN LIMIT field

The REJECT TO OPEN LIMIT field contains the minimum time in 10 µs increments that the target port shall wait to establish a connection request with an initiator port on an I_T nexus after receiving an OPEN_REJECT (RETRY), OPEN_REJECT (RESERVED CONTINUE 0), or OPEN_REJECT (RESERVED CONTINUE 1). This value may be rounded as defined in 2.3. A REJECT TO OPEN LIMIT field set to 0000h indicates that the minimum time is vendor specific. This minimum time is enforced by the port layer (see SAS-2).

4.3.17.3.4 Protocol-Specific Port mode page - Phy Control And Discover subpage (19h)

The Phy Control And Discover subpage contains phy-specific parameters. The mode page policy (see 4.4.10) for this subpage shall be shared. Parameters in this subpage shall affect only the referenced phy.

Table 308 defines the format of the subpage for SAS SSP.

Table 308. Protocol-Specific Port mode page SAS SSP - Phy Control And Discover subpage

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1B)	PAGE CODE (19h)					
1	SUBPAGE CODE (01h)							
2	(MSB)	PAGE LENGTH (N - 3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	GENERATION CODE							
7	NUMBER OF PHYS							
SAS phy mode descriptor list								
8	FIRST SAS PHY MODE DESCRIPTOR (SEE TABLE 309)							
...	...							
	LAST SAS PHY MODE DESCRIPTOR (SEE TABLE 309)							
n								

The PARAMETERS SAVEABLE (PS) bit is defined in 4.3.5.

The SPF field (see 4.3.5) shall be set to one to access the long format mode pages.

The PAGE CODE field (see 4.3.5) shall be set to 19h.

The SUBPAGE CODE field (see 4.3.5) shall be set to 01h.

The PAGE LENGTH field (see 4.3.5) shall be set to (4 + (the value of the NUMBER OF PHYS field) × (the length in bytes of the SAS phy mode descriptor)).

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field shall be set to 6h indicating this is a SAS SSP specific mode page.

GENERATION CODE field

The GENERATION CODE field is a one-byte counter that shall be incremented by one by the device server every time the values in this mode page or the Enhanced Phy Control mode page (see 4.3.17.3.6) are changed. A GENERATION CODE field set to 00h indicates the generation code is unknown. The device server shall wrap this field to 01h as the next increment after reaching its maximum value (i.e., FFh). The GENERATION CODE field is also contained in the Enhanced Phy Control mode page (see 4.3.17.3.6) and the Protocol-Specific Port log page (see SAS-2) and may be used to correlate phy settings across mode page and log page accesses.

Note. Device servers compliant with previous versions of this standard set the GENERATION CODE field to 00h.

NUMBER OF PHYS field

The NUMBER OF PHYS field contains the number of phys in the SAS target device and indicates the number of SAS phy mode descriptors that follow. This field shall not be changeable with MODE SELECT.

A SAS phy mode descriptor shall be included for each phy in the SAS target device (not just the SAS target port), starting with the lowest numbered phy and ending with the highest numbered phy.

Table 309 defines the SAS phy mode descriptor.

Table 309. SAS phy mode descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2 - 3	Reserved							
4	Reserved	ATTACHED DEVICE TYPE			Reserved			
5	REASON				NEGOTIATED PHYSICAL LINK RATE			
6	Reserved				ATTACHED SSP INITIATOR PORT	ATTACHED STP INITIATOR PORT	ATTACHED SMP INITIATOR PORT	Reserved
7	Reserved				ATTACHED SSP TARGET PORT	ATTACHED STP TARGET PORT	ATTACHED SMP TARGET PORT	Reserved
8 - 15	SAS ADDRESS							
16 - 23	ATTACHED SAS ADDRESS							
24	ATTACHED PHY IDENTIFIER							

Bit Byte	7	6	5	4	3	2	1	0
25 31	Reserved							
32	PROGRAMMED MINIMUM PHYSICAL LINK RATE				HARDWARE MINIMUM PHYSICAL LINK RATE			
33	PROGRAMMED MAXIMUM PHYSICAL LINK RATE				HARDWARE MAXIMUM PHYSICAL LINK RATE			
34 41	Reserved							
42 43	Vendor specific							
44 47	Reserved							

The PHY IDENTIFIER field, ATTACHED DEVICE TYPE field, NEGOTIATED PHYSICAL LINK RATE field, ATTACHED SSP INITIATOR PORT bit, ATTACHED STP INITIATOR PORT bit, ATTACHED SMP INITIATOR PORT bit, ATTACHED SSP TARGET PORT bit, ATTACHED STP TARGET PORT bit, ATTACHED SMP TARGET PORT bit, SAS ADDRESS field, ATTACHED SAS ADDRESS field, ATTACHED PHY IDENTIFIER, HARDWARE MINIMUM PHYSICAL LINK RATE field, and HARDWARE MAXIMUM PHYSICAL LINK RATE are defined in SAS-2.

MAXIMUM PHYSICAL LINK RATE field

MAXIMUM PHYSICAL LINK RATE field are defined in the SMP DISCOVER function (see SAS-2). These fields shall not be changeable with MODE SELECT.

PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field

The PROGRAMMED MINIMUM PHYSICAL LINK RATE field and PROGRAMMED MAXIMUM PHYSICAL LINK RATE field are defined in the SMP PHY CONTROL function (see SAS-2).

4.3.17.3.5 Port Control mode page

The Port Control Mode Page contains parameters that affect SSP target port operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT or MODE SENSE commands.

The mode page policy (see 4.4.10) for this mode page shall be shared.

Table 310 defines the format of this mode page.

Table 310. Port Control Mode Page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (19h)					
1	SUBPAGE CODE (02h)							
2	(MSB)	PAGE LENGTH (000Ch)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	(MSB)	POWER LOSS TIMEOUT						
7								(LSB)
8	Reserved							
15								

The Parameters Saveable (PS) bit is defined in 4.3.5.

The SubPage Format (SPF) bit is defined in 4.3.5 and shall be set to the value defined in table 310.

The PAGE CODE field is defined in 4.3.5 and shall be set to the value defined in table 310.

The SUBPAGE CODE field is defined in 4.3.5 and shall be set to the value defined in table 310.

The PAGE LENGTH field is defined in 4.3.5 and shall be set to the value defined in table 310.

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field is defined in 4.3.5 and shall be set to the value defined in table 310 indicating that this is a SAS SSP specific mode page.

POWER LOSS TIMEOUT field

The POWER LOSS TIMEOUT field contains the maximum time, in one millisecond increments, that a target port shall respond to connection requests with OPEN_REJECT (RETRY) after receiving NOTIFY (POWER LOSS EXPECTED) (see SAS-2). A POWER LOSS TIMEOUT field set to 0000h specifies that the maximum time is vendor-specific. The power loss timeout shall be restarted on each NOTIFY (POWER LOSS EXPECTED) that is received.

4.3.17.3.6 Enhanced Phy Control mode page

The Enhanced Phy Control mode page contains parameters that affect SSP target phy operation. If the mode page is implemented by one logical unit in a SCSI target device, then it shall be implemented by all logical units in the SCSI target device that support the MODE SELECT or MODE SENSE commands.

The mode page policy (see 4.4.10) for this mode page shall be shared.

Table 311 defines the format of this mode page.

Table 311. Enhanced Phy Control mode page

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF(1b)	PAGE CODE (19h)					
1	SUBPAGE CODE (03h)							
2	(MSB)	PAGE LENGTH (n - 3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	GENERATION CODE							
7	NUMBER OF PHYS							
Enhanced phy control mode descriptor list								
8	Enhanced phy control mode descriptor (first) (see table 312)							
27								
...	...							
n - 19	Enhanced phy control mode descriptor (last) (see table 312)							
n								

The Parameters Saveable (PS) bit is defined in 4.3.5.

The SubPage Format (SPF) bit is defined in 4.3.5 and shall be set to the value defined in table 311.

The PAGE CODE field is defined in 4.3.5 and shall be set to the value defined in table 311.

The SUBPAGE CODE field is defined in 4.3.5 and shall be set to the value defined in table 311.

The PAGE LENGTH field is defined in 4.3.5 and shall be set to the value defined in table 311 (i.e., 4 + (the value of the NUMBER OF PHYS field) × (the length in bytes of the SAS phy mode descriptor)).

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field is defined in table 290 and shall be set to the value defined in table 311 indicating that this is a SAS SSP specific mode page.

GENERATION CODE field

The GENERATION CODE field is defined in the Phy Control and Discover mode page (see 4.3.17.3.4).

The NUMBER OF PHYS field contains the number of phys in the SAS target device and indicates the number of enhanced phy control mode descriptors in the enhanced phy control mode descriptor list. This field shall not be changeable with the MODE SELECT command.

The enhanced phy control mode descriptor list contains an enhanced phy control mode descriptor for each phy in the SAS target device, not just the SAS target port, starting with the lowest numbered phy and ending with the highest numbered phy.

Table 312 defines the enhanced phy control mode descriptor.

Table 312. Enhanced phy control mode descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	PHY IDENTIFIER							
2	(MSB)	DESCRIPTOR LENGTH (0010h)						
3								(LSB)
4	PROGRAMMED PHY CAPABILITIES							
7								
8	CURRENT PHY CAPABILITIES							
11								
12	ATTACHED PHY CAPABILITIES							
15								
16	Reserved							
17								
18	Reserved			NEGOTIATED SSC	NEGOTIATED PHYSICAL LINK RATE			
19	Reserved							HARDWARE MUXING SUPPORTED

DESCRIPTOR LENGTH field

The DESCRIPTOR LENGTH field contains the length in bytes that follow in the descriptor and shall be set to the value defined in table 312.

The fields in the enhanced phy control mode descriptor not defined in this subclause are defined in the SMP DISCOVER response (see SAS-2). These fields shall not be changeable with the MODE SELECT command.

4.3.17.3.7 Protocol-Specific Port – Transceiver Control (Out) subpage

The Transceiver Control (Out) subpage (see table 313) of the Protocol-Specific Port mode page is used to adjust the transceiver parameter values of a phy. This subpage, when implemented, must comply with the following additional itemized requirements:

1) The transceiver parameter settings may only be adjusted temporarily using the contents of this mode page, and cannot be saved.
2) New phy settings are not to take effect until after status is returned for the MODE SELECT command that is issued to adjust the phy settings.
3) The Transceiver Control subpage shall not be returned for a MODE SENSE command to return all pages. It shall only be returned to a MODE SENSE command, specifically, for the Transceiver Control subpage)..

Table 313. Transceiver Control (Out) subpage

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF(1b)	PAGE CODE (19h)					
1	SUBPAGE CODE (E5h)							
2	(MSB)	PAGE LENGTH (n - 3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	NUMBER OF PHYS (02h for a dual ported disk drive)							
7	NUMBER OF PHYS							
8 - 14	Reserved							
15	TRANSCEIVER CONTROLS MODE DESCRIPTOR LENGTH							
16	First phy Transceiver Controls mode descriptor							
...		...						
n	Last phy Transceiver Controls mode descriptor							

Transceiver Controls mode descriptors

Detail for the mode descriptor(s) for transceiver controls can be found in Transceiver Controls descriptor table 314. All supported link rates and controls shall be listed.

Table 314. Transceiver Controls descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	PHY IDENTIFIER							
1	CONTROLLABLE LINK RATE							
2	DRIVER STRENGTH CONTROL							
3	DRIVER EMPHASIS CONTROL							
4	DRIVER SLEW RATE CONTROL							
5 8	MANUFACTURER DEFINE BYTES FOR TRANSMITTER CONTROL							
9	RECEIVER THRESHOLD CONTROL							
10	RECEIVER EQUALIZATION GAIN CONTROL							
11 13	MANUFACTURER DEFINE BYTES FOR RECEIVER CONTROL							
14	Reserved			DTS	DAG	Reserved		
15	Reserved							

PHY IDENTIFIER field

SAS devices may have more than one target port. If a target device has 2 operational target ports (2 phys - primary and secondary port), then this field shall be set to either 0h for the primary port or 1h for the secondary port. This field is unchangeable.

Controllable Link Rate field

The Controllable Link Rate field shall contain the controllable link rate for the phy. Table 315 lists the supported values Controllable Link Rates. This field is not changeable.

Table 315. Controllable link rates

Value	Definition
0x00	Phy exists, but does not support “Transceiver Control” controls
0x01 – 0x07	Reserved
0x08	Phy supports Transceiver Control controls for 1.5 Gbps
0x09	Phy supports Transceiver Control controls for 3.0 Gbps
0x0A	Phy supports Transceiver Control controls for 6.0 Gbps
0x0B – 0xFF	Reserved

Driver Strength Control field

The Driver Strength Control field shall contain either the user input (MODE SELECT) or current (MODE SENSE) driver strength output setting of the target device SAS phy. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command. The Driver Strength Control field shall be supported as described in table 316. This field is changeable but not savable.

Table 316. Driver strength control

Value	Definition
0x66 (write only)	Write 66h shall reset the driver strength setting back to factory default value.
0x00 – Range Limit (write only)	User input for driver strength control setting. If the new user input is out of range specified in subpage 19h/E6h, the target device shall maintain the current setting.
Read Only	The current setting value shall be displayed.

DRIVER EMPHASIS CONTROL field

The DRIVER EMPHASIS CONTROL field shall contain either the user input (MODE SELECT) or current (MODE SENSE) driver emphasis setting of the target device SAS phy. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command. The DRIVER EMPHASIS CONTROL field shall be supported as described in table 317. This field is changeable but not savable.

Table 317. Driver emphasis control

Value	Definition
0x66 (write only)	Write 66h shall reset the driver pre-emphasis setting back to factory default value.
0x00 – Range Limit (write only)	User input for driver pre-emphasis control setting. If the new user input is out of range specified in subpage 19h/E6h, the target device shall maintain the current setting.
Read Only	The current setting value shall be displayed.

DRIVER SLEW RATE CONTROL field

The DRIVER SLEW RATE CONTROL field shall contain either the user input (MODE SELECT) or current (MODE SENSE) driver slew rate setting of the target device SAS phy. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command. The DRIVER SLEW RATE CONTROL field shall be supported as described in table 318. This field is changeable but not savable.

Table 318. Driver slew rate control

Value	Definition
0x66 (write only)	Write 66h shall reset the driver slew rate setting back to factory default value.
0x00 – Range Limit (write only)	User input for driver slew rate control setting. If the new user input is out of range specified in subpage 19h/E6h, the target device shall maintain the current setting.
Read Only	The current setting value shall be displayed.

MANUFACTURER DEFINED BYTES FOR TRANSMITTER CONTROL field

The MANUFACTURER DEFINED BYTES FOR TRANSMITTER CONTROL field may be used by the manufacturer who needs addition control bytes for the transmitter. All control bytes shall accept 66h as reset code which change the control byte back to factory default setting. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command.

RECEIVER THRESHOLD CONTROL field

The RECEIVER THRESHOLD CONTROL field contains either the user input (MODE SELECT) or current (MODE SENSE) receiver threshold setting of the target device SAS phy. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command. The RECEIVER THRESHOLD CONTROL field shall be supported as described in table 319. This field is changeable but not savable.

Table 319. Receiver threshold control

Value	Definition
0x66 (write only)	Write 66h shall reset the receiver threshold setting back to factory default value.
0x00 – Range Limit (write only)	User input for receiver threshold control setting. If the new user input is out of range specified in subpage 19h/E6h, the target device shall maintain the current setting.
Read Only	The current setting value shall be displayed.

RECEIVER EQUALIZATION GAIN CONTROL field

The RECEIVER EQUALIZATION GAIN CONTROL field contains either the user input (MODE SELECT) or current (MODE SENSE) receiver equalization gain setting of the target device SAS phy. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command. This field shall be supported as described in table 320. This field is changeable but not savable.

Table 320. Receiver equalization gain control field

Value	Definition
0x66 (write only)	Write 66h shall reset the receiver equalization gain setting back to factory default value for the driver voltage.
0x00 – Range Limit (write only)	User input for receiver equalization gain control setting. If the new user input is out of range specified in subpage 19h/E6h, the target device shall maintain the current setting
Read Only	The current setting value shall be displayed.

MANUFACTURER DEFINED BYTES FOR RECEIVER CONTROL field

The MANUFACTURER DEFINED BYTES FOR RECEIVER CONTROL field may be used by manufacturer who needs addition control bytes for the receiver. All control bytes shall accept 66h as reset code which change the control byte back to factory default setting. If the user input setting is out of range of the transceiver capability, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command.

DAG (Disable ALIGN Generation) bit

The DAG (Disable ALIGN Generation) bit is the control for the ALIGN primitive generation by the target device phy.

0 If the DAG bit is set to 0b, the target device phy shall insert the necessary ALIGN primitives as defined in the SAS specification. If DAG is set to 1b, then the target device phy shall suppress all ALIGN primitive insertion.

1 If the DAG bit is to 1b and the target device does not support the DAG bit, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command (the DAG field shall not be updated).

This field is changeable but not savable.

DTS (Disable Transmitter Scrambling) bit

The DTS (Disable Transmitter Scrambling) bit controls the transmitter data scrambling by the target device phy.

0 If DTS is set to 0b, the target device phy shall transmit data scrambled as defined in the SAS specification. If DTS is set to 1b, the target device shall disable the transmitter data scrambling.

1 If the DTS field is set to 1b and the target device does not support the DTS field, the target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for the MODE SELECT command (the DTS field shall not be updated).

The DTS bit is changeable but not savable.

4.3.17.3.8 Protocol-Specific Port – Transceiver Control (In) subpage

The Transceiver Control (In) subpage of the Protocol-Specific Port mode page (see table 321) is used to return the transceiver parameter values of a phy. The Transceiver Control subpage shall not be returned for a MODE SENSE command to return all pages. It shall only be returned to a MODE SENSE command, specifically, for the Transceiver Control subpage. The target device shall return a CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST for a MODE SELECT command issued for this subpage.

Table 321. Protocol-Specific Port - Transceiver Control (In) subpage

Bit / Byte	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	Page Code 19h					
1	SUBPAGE CODE (E6h)							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
4	Reserved							
5	Reserved				PROTOCOL IDENTIFIER (6h)			
6	Reserved							
7	NUMBER OF PHYS (02h for a dual ported disk drive)							
8	Reserved							
14								
15	Transceiver Controls mode descriptor length							
16	First phy Transceiver Controls mode descriptor							
•••	•••							
n	Last phy Transceiver Controls mode descriptor							

Number of Phys field

SAS devices may have more than one target port. If a target device has 2 operational target ports (2 phys), then this field shall be set to 2h. Target devices with only one port shall set this field to 1h. The field is unchangeable.

Transceiver Controls MODE DESCRIPTOR Length field

This “length” field shall contain the total length in bytes of each phy’s transceiver controls mode descriptor. Since the descriptors are the same length, this value is not provided in each descriptor, but rather, once in this subpage prior to the descriptor(s). The field is unchangeable.

4.3.17.3.9 Transceiver Controls mode descriptors

Detail for the mode descriptor(s) for transceiver controls can be found in table 322.

Table 322. Transceiver Controls descriptor

Bit / Byte	7	6	5	4	3	2	1	0
0	PHY IDENTIFIER							
1	CONTROLLABLE LINK RATE							
2	MAX DRIVER STRENGTH CONTROL							
3	MAX DRIVER EMPHASIS CONTROL							
4	MAX DRIVER SLEW RATE CONTROL							
5	DRIVER STRENGTH CONTROL CHANGE PER STEP							
6	DRIVER EMPHASIS CONTROL CHANGE PER STEP							
7	DRIVER SLEW RATE CONTROL CHANGE PER STEP							
8	DDh							
9-20	MANUFACTURER DEFINED BYTES INFO FOR TRANSMITTER CONTROL							
21	MAX RECEIVER THRESHOLD CONTROL							
22	MAX RECEIVER EQUALIZATION GAIN CONTROL							
23	RECEIVER THRESHOLD CONTROL CHANGE PER STEP							
24	RECEIVER EQUALIZATION GAIN CONTROL CHANGE PER STEP							
25-32	MANUFACTURER DEFINED BYTES INFO FOR RECEIVER CONTROL							

PHY IDENTIFIER field

SAS devices may have more than one target port. If a target device has 2 operational target ports (2 phys - primary and secondary port), then this field shall be set to either 0h for the primary port or 1h for the secondary port. The field is unchangeable.

Controllable Link Rate field

This field shall contain the controllable link rate for the phy. Table 323 lists the supported values for the field. This field is not changeable.

Table 323. Controllable link rates field

Value	Definition
0x00	Phy exists, but does not support Transceiver Control controls
0x01 – 0x07	Reserved
0x08	Phy supports Transceiver Control controls for 1.5 Gbps
0x09	Phy supports Transceiver Control controls for 3.0 Gpbs
0x0A	Phy supports Transceiver Control controls for 6.0 Gbps
0x0B – 0xFF	Reserved

Max Driver Strength Control field

This field shall contain the max allowable driver strength output setting of the target device SAS phy. The field is unchangeable.

Max Driver Emphasis Control field

This field shall contain the max allowable driver emphasis setting of the target device SAS phy. The field is unchangeable.

Max Driver Slew Rate Control field

This field shall contain the max allowable driver slew rate setting of the target device SAS phy. The field is unchangeable.

Driver Strength Control Change Per Step

This field shall contain the approximate voltage change (in mV unit) for each increment or decrement of driver strength control setting. The field is unchangeable.

Driver Emphasis Control Change Per Step field

This field shall contain the approximate percentage of amplitude change (in% unit) for each increment or decrement of driver emphasis control setting. This field shall be used to establish the transceiver driver pre-emphasis or de-emphasis. The field value should be treated as a 2's-complement signed value that can range from 1 to +100 for pre-emphasis, or -1 to -100 for de-emphasis. The field is unchangeable.

Driver Slew Rate Control Change Per Step field

This field shall contain the approximate slew rate change (in mV/nsec unit) for each increment or decrement of driver slew rate control setting. The field is unchangeable.

Max Receiver Threshold Control field

The Max Receiver Threshold Control field contains max allowable threshold setting of the target device SAS phy. This field is unchangeable.

MANUFACTURER DEFINED BYTES INFO FOR TRANSMITTER CONTROL field

The MANUFACTURER DEFINED BYTES INFO FOR TRANSMITTER CONTROL field can be used by the manufacturer who needs addition control bytes for the transmitter.

Max Receiver Equalization Gain Control field

The Max Receiver Equalization Gain Control field contains the max allowable receiver equalization gain setting of the target device SAS phy. This field is unchangeable.

Receiver Threshold Control Change Per Step field

The Receiver Threshold Control Change Per Step field shall contain the approximate voltage change (in mV unit) for each increment or decrement of receiver threshold control setting. This field is unchangeable.

Receiver Equalization Gain Control Change Per Step field

The Receiver Equalization Gain Control Change Per Step field shall contain the approximate gain change for each increment or decrement of receiver equalization gain control setting. This field is unchangeable.

MANUFACTURER DEFINED BYTES INFO FOR RECEIVER CONTROL field

The MANUFACTURER DEFINED BYTES INFO FOR RECEIVER CONTROL field can be used by the manufacturer who needs addition control bytes for the receiver.

4.3.18 Rigid Drive Geometry Parameters page (04h)

This mode page has been declared obsolete by the T10 committee. However, it is included because it may be implemented on some products.

The Rigid Drive Geometry Parameters page implementation is defined in Table 324. This table summarizes the function and defines the default value.

Table 324. Rigid Drive Geometry Parameters page (04h)

Bit Byte	7	6	5	4	3	2	1	0
				Page Descriptor Header				
0	PS	Reserved	PAGE CODE (04h)					
1	PAGE LENGTH (16h)							
				Rigid Drive Geometry Parameters				
2	NUMBER OF CYLINDERS (MSB)							
3	NUMBER OF CYLINDERS							
4	NUMBER OF CYLINDERS (LSB)							
5	NUMBER OF HEADS							
6,7,8	STARTING CYLINDER—WRITE PRECOMP							
9,10,11	STARTING CYLINDER—REDUCED WRITE CURRENT							
12,13	DRIVE STEP RATE							
14,15,16	LANDING ZONE CYLINDER							
17	0	0	0	0	0	0	RPL	
18	ROTATIONAL OFFSET XXh							
19	Reserved							
20	(MSB)	MEDIUM ROTATION RATE						
21								(LSB)
22	Reserved							
23	Reserved							

PS (Parameter Savable) bit

The Parameter Savable (PS) bit of 1 indicates that page 04 parameter data is savable and is saved when a Format Function is performed. In some drives an exception exists that applies to bytes 17 and 18. In the exception drives, bytes 17 and 18 are only saved if the SMP bit in the MODE SELECT command (Table 62) is 1. See individual drive's Product Manual, Volume 1, SCSI bus conditions and miscellaneous features supported." This PS bit is not applicable to the MODE SELECT command.

NUMBER OF CYLINDERS field

The NUMBER OF CYLINDERS field defines the number of physical cylinders used for data storage. This may or may not include spare cylinders set aside for flaw reallocation. See individual drive's Product Manual, Volume 1, which specifies what the drive reports.

The drive uses some additional cylinders for storing drive parameters, defect lists, or for diagnostic purposes. These are not accessible by the user.

NUMBER OF HEADS field

The NUMBER OF HEADS field indicates the maximum number of data (read/write) heads on the drive.

STARTING CYLINDER—WRITE PRECOMP, STARTING CYLINDER—REDUCED WRITE CURRENT, DRIVE STEP RATE, and LANDING ZONE CYLINDER fields

Not applicable.

RPL (rotational position locking) field

Used for Spindle Synchronization (rotational position locking) (see table 325).

Table 325. Rotational Position Locking

RPL	Description
00b	Indicates that spindle synchronization is automatic. (Automatic master arbitration is used to determine which device in the chain is to be master.)
01b	The target operates as a synchronized-spindle slave.
10b	The target operates as a synchronized-spindle master.
11b	The target operates as a synchronized-spindle master control (not supported by drive).

ROTATIONAL OFFSET field

Rotational skew in the lagging direction used for spindle synchronization. The value XXh given represents a XXh/FFh fractional part of a revolution lagging offset. One revolution lag is maximum. See clause on Synchronous Spindle Operation in individual drive's Product Manual, Volume 1.

MEDIUM ROTATIORN RATE field

On MODE SENSE command, these bytes return drive nominal rotation rate in revolutions per minute for synchronous spindle operation. The bytes have no meaning for MODE SELECT.

See individual drive's Product Manual, Volume 1, MODE SENSE Data clause, for changeable values.

4.3.19 Unit Attention parameters page (00h)

The Unit Attention parameters page is the last page to be reported by the drive.

Table 326. Unit Attention parameters page (00h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PS		PAGE CODE (00h)					
1	PAGE LENGTH (in bytes)							
2	PM	SSM	INQUIRY LENGTH	UNIT ATTENTION	DFUA	ROUND	STRICT	SCSI-2
3	DAR	SELF SEEK	Reserved					
4	VJIT Disabled	Reserved			JIT3	JIT2	JIT1	JIT0
5	Reserved							
6	Reserved							
7	Reserved							

PS (Parameter Savable) bit

A Parameter Savable (PS) bit of one indicates that the drive is capable of saving the page in a nonvolatile vendor-specific location (used only with MODE SENSE command).

PAGE LENGTH field

The PAGE LENGTH field specifies the length in bytes of the mode parameters that follow. If the initiator does not set this value to the value that is returned for the page by the MODE SENSE command, the drive shall terminate the command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to Invalid Field In Parameter List. The drive is permitted to implement a mode page that is less than the full page length defined by this specification, provided no field is truncated and the page length field correctly specifies the actual length implemented. If the STRICT bit equals zero and if the page length specified by the initiator is shorter than the actual page length, then the parameters are transferred and the command ends with GOOD status if no other items cause the command to be rejected.

Caution: Utilization of this forgiving option by a SCSI initiator port that does not analyze the impact of the truncation could adversely affect data integrity.

PM (Performance Mode) bit

The PM (Performance Mode) bit is used to control the drive's cache management algorithm to allow best performance in different types of systems. It is the initiator's responsibility to determine which setting is best for that system.

1 A PM value of 1 indicates that the number of cache segments is fixed to the value set in mode page 08h.

0 A PM value of 0 indicates that the drive will optimize the number of segments depending on the command activity observed by the drive. The number of segments value (in mode page 08h) is ignored with the PM value is 0.

SSM (enable Synchronous Select Mode) bit

1 If the enable Synchronous Select Mode (SSM) bit equals one, the drive initiates WDTR and SDTR messages when it recognizes that one may be required (after reset, reset message, or power cycle).

0 If the SSM bit equals zero, the drive does not initiate WDTR or SDTR regardless of negotiated conditions prior to reset, reset message, or power cycle.

IL (INQUIRY Length) bit

1 When the INQUIRY Length (IL) bit is set to 1, the standard INQUIRY data available to a host is limited to the 36 bytes required by the SCSI-2 specification.

0 When the IL bit is reset (0), 148 bytes of standard INQUIRY data are available. The Additional Length field in byte 4 of the INQUIRY data is updated to reflect the actual number of additional bytes available.

UNIT ATTENTION bit

1 When the UNIT ATTENTION bit is set to 1, then UNIT ATTENTION is logged in sense only; no CHECK CONDITION status is presented following any reset.

0 When this bit is Reset (0), then Check Condition is presented for all affected initiators following a reset until REQUEST SENSE is issued by each initiator (as per current operation).

DFUA (Disable Force Unit Access) bit

1 When the Disable Force Unit Access (DFUA) bit is set to 1, the drive ignores the FUA bit in READ and WRITE commands. This can result in better drive performance in some circumstances.

0 When the DFUA is set to 0, the drive obeys the FUA setting in READ and WRITE commands.

ROUND bit

1 When the ROUND bit equals one, the drive treats and reports rounded parameters as described in 2.3.

0 When ROUND equals zero, the drive shall round the parameter and handle command completion reporting as if the parameter had not been rounded.

STRICT bit

1 When the STRICT bit is a one, the drive checks for initiator attempts to change unchangeable parameters. If the drive detects an attempt, it rejects the command in the standard way, i.e., CHECK CONDITION status from drive, REQUEST SENSE from the initiator, and Illegal Request Sense key (5h) back from the drive.

0 When the STRICT bit is zero, the drive ignores the values of the unchangeable parameters in a MODE SELECT command. The drive does not reject the command trying to change unchangeable parameters.

SCSI-2 bit

1 When set to one, the SCSI-2 bit changes the following SCSI-3 features from their SCSI-3 definition to the SCSI-2 definition.

0 When S2 equals zero, the following features remain as specified in other portions of this specification:

a) Control Mode Page (0Ah) Length from 0Ah to 06h.

b) Caching Page (08h) Length from 12h to 0Ah.

DAR (Deferred Auto Reallocation) bit

1 Deferred reallocation is enabled. When an unrecoverable read error occurs, the drive remembers the LBA of the error; when a subsequent write command occurs for that LBA and the AWRE bit in mode page 1 is enabled, then the LBA is reallocated before the write command is completed.

0 Deferred reallocation is disabled.

Self Seek bit

1 If the Self Seek bit is set to one, the drive will enter self seek mode for testing purposes. Such testing could include, but is not limited to, power dissipation and acoustics. While in this mode, the drive will accept SCSI commands and will process them in between the self seek operations, including a MODE SELECT to turn this bit back off. As such, this bit should be off for normal drive operations.

0 If this bit is set to zero, the drive will not self seek; normal operating mode.

VJIT Disabled bit

1 When this bit is set to 1, the drive will always use the fastest seek type (i.e., JIT 0). JIT bits 0:3 of byte 4 are ignored.

0 When this bit is set to 0, the drive will operate according to the settings of JIT bits 0:3.

JIT (Just in Time) bits

The four JIT (Just In Time) bits allow you to enable and disable certain seek speeds. JIT0 represents the fastest seek type used by the drive, JIT1 represents the second fastest, JIT2 represents the third fastest, and JIT3 represents the slowest seek type. You can use these bits to reduce acoustics by disabling the fastest seeks. This can also reduce power consumption (from seek activity). These JIT settings only affect user read and write operations. Background drive operations and user seek commands will always use the fastest seek type. When the bit is set to 1, the drive is allowed to use this seek type and any slower seek types in its seek speed algorithm. When the bit is set to 0, the drive is not allowed to use this seek type in its seek speed algorithm unless a faster seek type is selected. If all JIT bits are set to zero or if all bits are set to one, the drive can use any of the seek speeds.

Reserved bits

These bits are reserved for future compatibility with Seagate host adapters. Though they presently may be changeable, these bits do not control anything, unless the individual drive's Product Manual, Volume 1, indicates that they do and defines their use in the MODE SENSE Data section.

Reserved bytes

See individual drive's Product Manual, Volume 1, MODE SENSE Data section for a table showing codes that indicate which of these bits are changeable by the host using the MODE SELECT command.

4.3.20 Verify Error Recovery mode page (07h)

The Verify Error Recovery mode page (see table 327) specifies the error recovery parameters the device server shall use during the VERIFY command and the verify operation of the WRITE AND VERIFY command.

Table 327. Verify Error Recovery mode page

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (07h)					
1	PAGE LENGTH (0Ah)							
2	Reserved				ERROR RECOVERY BITS			
	AWRE	ARRE	TB	RC	EER	PER	DTE	DCR
3	VERIFY RETRY COUNT							
4	Obsolete formerly CORRECTION SPAN							
5		Reserved formerly HEAD OFFSET COUNT (Byte 5), DATA STROBE OFFSET COUNT (Byte 6) and WRITE RETRY COUNT (Byte 8) Bytes 7 and 9 were Reserved						
9								
10	(MSB)	VERIFY RECOVERY TIME LIMIT						
11								(LSB)

The parameters saveable (PS) bit, the subpage format (SPF) bit, the PAGE CODE field, the SUBPAGE CODE field, and the PAGE LENGTH field are defined in 4.3.4.

AWRE bit

The AWRE bit as defined in the Read-Write Error Recovery mode page (see 4.3.20) applies to the WRITE AND VERIFY command. The VERIFY command shall not perform automatic reallocation.

EER bit, PER bit, DTE bit, and DCR bit

The EER bit, the PER bit, the DTE bit, and the DCR bit (i.e., the error recovery bits) are defined in 4.3.7. The combinations of these bits are defined in table 270 (see 4.3.7).

VERIFY RETRY COUNT field

The VERIFY RETRY COUNT field specifies the number of times that the device server shall attempt its recovery algorithm during a verify operation.

VERIFY RECOVERY TIME LIMIT field

The VERIFY RECOVERY TIME LIMIT field specifies in milliseconds the maximum time duration that the device server shall use error recovery procedures to recover data for an individual logical block. The device server may round this value as described in 2.3.

When both a retry count and a recovery time limit are specified, the one that requires the least time for data error recovery actions shall have priority.

To disable all types of correction and retries the application client should set the EER bit to zero, the PER bit to one, the DTE bit to one, the DCR bit to one, the VERIFY RETRY COUNT field to 00h, and the VERIFY RECOVERY TIME LIMIT field to 0000h.

4.4 Vital product data parameters

4.4.1 Vital product data parameters overview and page codes

This subclause describes the vital product data (VPD) page structure and the VPD pages (see table 328) that are applicable to all SCSI devices. These VPD pages are returned by an INQUIRY command with the EVPD bit set to one (see 3.6) and contain vendor specific product information about a logical unit and SCSI target device. The vital product data may include vendor identification, product identification, unit serial numbers, device operating definitions, manufacturing data, field replaceable unit information, and other vendor specific information.

Table 328. Vital product data page codes

Page code	VPD Page Name	Reference	Support Requirements
01h - 7Fh	ASCII Information	4.4.2	Optional
83h	Device Identification	4.4.8	Mandatory
86h	Extended INQUIRY Data		Optional
85h	Management Network Addresses		Optional
87h	Mode Page Policy	4.4.10	Optional
81h	Obsolete		
82h	Obsolete		
88h	SCSI Ports		Optional
84h	Software Interface Identification		Optional
00h	Supported VPD Pages	4.4.11	Mandatory
80h	Unit Serial Number	4.4.12	Optional
89h - AFh	Reserved		
B0F	Block Limits	4.4.3	Optional
B1h - BFh	(See specific device type)		
C0h - FFh	Vendor specific	4.4.4 - 4.4.9	

4.4.2 ASCII Information VPD page (01h - 7Fh)

The ASCII Information VPD page (see table 329) contains information for the field replaceable unit code returned in the sense data (see 2.2.9).

Table 329. ASCII Information VPD page

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (01h - 7Fh)							
2	Reserved							
3	PAGE LENGTH (N-3)							
4	ASCII LENGTH (M-4)							
5	(MSB)	ASCII INFORMATION						
m								(LSB)
m+1	Vendor specific information							
n								

PERIPHERAL QUALIFIER and the PERIPHERAL DEVICE TYPE fields

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field contains the same value as in the PAGE OR OPERATION CODE field of the INQUIRY CDB (see 3.6) and is associated with the FIELD REPLACEABLE UNIT CODE field returned in the sense data.

Note. The FIELD REPLACEABLE UNIT CODE field in the sense data provides for 255 possible codes, while the PAGE CODE field provides for only 127 possible codes. For that reason it is not possible to return ASCII Information VPD pages for the upper code values.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

ASCII LENGTH field

The ASCII LENGTH field specifies the length in bytes of the ASCII INFORMATION field that follows. A value of zero in this field indicates that no ASCII information is available for the specified page code. The relationship between the ASCII LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

The ASCII INFORMATION field contains ASCII information concerning the field replaceable unit identified by the page code. The data in this field shall be formatted in one or more character string lines. Each line shall contain only graphic codes (i.e., code values 20h through 7Eh) and shall be terminated with a NULL (00h) character.

The contents of the vendor specific information field is not defined in this manual.

4.4.3 Block Limits VPD page

The Block Limits VPD page (see table 330) provides the application client with the means to obtain certain operating parameters of the logical unit.

Table 330. Block Limits VPD page (Sheet 1 of 2)

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (B0h)							
2	Reserved							
3	PAGE LENGTH							
4	Reserved							
5								
6	(MSB)	OPTIMAL TRANSFER LENGTH GRANULARITY						
7								(LSB)
8	(MSB)	MAXIMUM TRANSFER LENGTH						
.....								
11								(LSB)
12	(MSB)	OPTIMAL TRANSFER LENGTH						
.....								
15								(LSB)
16	(MSB)	MAXIMUM PREFETCH XDREAD XDWRITE TRANSFER LENGTH						
.....								
19								(LSB)
20	(MSB)	MAXIMUM UNMAP LBA COUNT						
.....								
23								(LSB)
24	(MSB)	MAXIMUM UNMAP BLOCK DESCRIPTOR COUNT						
.....								
27								(LSB)
28	(MSB)	OPTIMAL UNMAP GRANDULARITY						
.....								
31								(LSB)

Table 330. Block Limits VPD page (Sheet 2 of 2)

Bit / Byte	7	6	5	4	3	2	1	0
28	UGAVALID	(MSB)						
.....		UNMAP GRANDULARITY ALIGNMENT						
31								(LSB)
32		Reserved						
.....								
64								

PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

The PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 330.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

If the device server supports thin provisioning, then the device server shall set the PAGE LENGTH field to 3Ch. If the device server does not support thin provisioning, then the device server may set the PAGE LENGTH field to 10h.

OPTIMAL TRANSFER LENGTH GRANULARITY field

The OPTIMAL TRANSFER LENGTH GRANULARITY field indicates the optimal transfer length granularity in blocks for a single ORWRITE command, PRE-FETCH command, READ command, VERIFY command, WRITE command, WRITE AND VERIFY command, XDREAD command, XDWRITE command, XDWRITEREAD command, or XPWRITE command. Transfers with transfer lengths not equal to a multiple of this value may incur significant delays in processing.

MAXIMUM TRANSFER LENGTH field

The MAXIMUM TRANSFER LENGTH field indicates the maximum transfer length in blocks that the device server accepts for a single ORWRITE command, READ command, VERIFY command, WRITE command, WRITE AND VERIFY command, XDWRITEREAD command, or XPWRITE command. Requests for transfer lengths exceeding this limit result in CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. A MAXIMUM TRANSFER LENGTH field set to zero indicates that there is no reported limit on the transfer length.

OPTIMAL TRANSFER LENGTH field

The OPTIMAL TRANSFER LENGTH field indicates the optimal transfer length in blocks for a single ORWRITE command, PRE-FETCH command, READ command, VERIFY command, WRITE command, WRITE AND VERIFY command, XDREAD command, XDWRITE command, XDWRITEREAD command, or XPWRITE command. Transfers with transfer lengths exceeding this value may incur significant delays in processing.

MAXIMUM PREFETCH XDREAD XDWRITE TRANSFER LENGTH field

The MAXIMUM PREFETCH XDREAD XDWRITE TRANSFER LENGTH field indicates:

a) the maximum transfer length in blocks that the device server accepts for a single PRE-FETCH command;
b) if the XOR Control mode page (see SBC-3) is implemented, then the maximum value supported by the MAXIMUM XOR WRITE SIZE field in the XOR Control mode page; and
c) if the XOR Control mode page is not implemented, then the maximum transfer length in blocks that the device server accepts for a single XDWRITE command or XDREAD command.

The device server should set the MAXIMUM PREFETCH XDREAD XDWRITE TRANSFER LENGTH field to less than or equal to the MAXIMUM TRANSFER LENGTH field.

MAXIMUM UNMAP LBA COUNT field

The MAXIMUM UNMAP BLOCK DESCRIPTOR COUNT field indicates the maximum number of LBAs that may be unmapped by an UNMAP command (see 3.56). If the number of LBAs that may be unmapped by an UNMAP command is constrained only by the amount of data that may be contained in the UNMAP parameter list (see 3.56.2), then the device server shall set the MAXIMUM UNMAP LBA COUNT field to FFFF_FFFFh. If the device server implements the UNMAP command, then the value in this field shall be greater than or equal to one. If the MAXIMUM UNMAP LBA COUNT field is set to 0000_0000h, or the PAGE LENGTH field is set to 10h, then the device server does not implement the UNMAP command.

MAXIMUM UNMAP BLOCK DESCRIPTOR COUNT field

The MAXIMUM UNMAP BLOCK DESCRIPTOR COUNT field indicates the maximum number of UNMAP block descriptors (see 3.56.2) that shall be contained in the parameter data transferred to the device server for an UNMAP command (see 3.56). If there is no limit on the number of UNMAP block descriptors contained in the parameter data, then the device server shall set the MAXIMUM UNMAP BLOCK DESCRIPTOR COUNT field to FFFF_FFFFh. If the device server implements the UNMAP command, then the value in this field shall be greater than or equal to one. If the device server implements the UNMAP command, then the value in this field shall be greater than or equal to one. If the MAXIMUM UNMAP BLOCK DESCRIPTOR COUNT field is set to 0000_0000h, or the PAGE LENGTH field is set to 10h, then the device server does not implement the UNMAP command.

OPTIMAL UNMAP GRANDULARITY field

The OPTIMAL UNMAP GRANDULARITY field indicates the optimal granularity in logical blocks for unmap requests (e.g., an UNMAP command or a WRITE SAME (16) command with the UNMAP bit set to one). An unmap request with a number of logical blocks that is not a multiple of this value may result in unmap operations on fewer LBAs than requested. If the OPTIMAL UNMAP GRANDULARITY field is set to 0000_0000h, then the optimal unmap granularity is not specified.

UGVALID (unmap granularity alignment valid) bit

An unmap granularity alignment valid (UGVALID) bit set to one indicates that the UNMAP GRANDULARITY ALIGNMENT field is valid. A UGVALID bit set to zero indicates that the UNMAP GRANDULARITY ALIGNMENT field is not valid.

UNMAP GRANDULARITY ALIGNMENT field

The UNMAP GRANDULARITY ALIGNMENT field indicates the LBA of the first logical block to which the OPTIMAL UNMAP GRANDULARITY field applies. The unmap granularity alignment is used to calculate an optimal unmap request starting LBA as follows:

optimal unmap request starting LBA = (n × optimal unmap granularity) + unmap granularity alignment

where:

n	is zero or any positive integer value
optimal unmap granularity	is the value in the OPTIMAL UNMAP GRANDULARITY field
unmap granularity alignment	is the value in the UNMAP GRANDULARITY ALIGNMENT field

An unmap request with a starting LBA that is not optimal may result in unmap operations on fewer LBAs than requested.

4.4.4 Date Code page (C1h)

Table 331. ETF Log Date Code page (C1h)

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER		PERIPHERAL DEVICE TYPE					
1	PAGE CODE (C1h)							
2	00h							
3	PAGE LENGTH (10h)							
4	(MSB)							
:	ETF LOG DATE IN ASCII							
11								(LSB)
12	(MSB)							
:	COMPILE DATE CODE							
19								(LSB)

PERIPHERAL QUALIFIER field and PERIPHERAL DEVICE TYPE field

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 331.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

ETF Log field

The ETF Log date code field contains ASCII data. The data is stored in the format MMDDYYYY. Leading ASCII zero characters are added to single-digit months or days.

COMPILE DATE CODE field

The COMPILE DATE CODE field contains 8 ASCII bytes of data for a date of the form MMDDYYYY.

4.4.5 Device Behavior page (C3h)

The Device Behavior page (see table 332) will be used by the regression tests to determine what behavior should be expected from a particular firmware package.

Table 332. Device Behavior page (C3h)

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (C3h)							
2	Reserved							
3	PAGE LENGTH							
4	VERSION NUMBER							
5	BEHAVIOR CODE							
6	BEHAVIOR CODE VERSION NUMBER							
7 : 22	ASCII FAMILY NUMBER (16 bytes)							
23	NUMBER OF INTERLEAVES							
24	DEFAULT NUMBER OF CACHE SEGMENTS							
25+	Feature Flags and Additional Byte Fields will go here but are undefined at this time.							

PERIPHERAL QUALIFIER field and PERIPHERAL DEVICE TYPE fields

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 332.

PAGE LENGTH field

PAGE LENGTH defines the length of the Device Behavior information in bytes. If the ALLOCATION LENGTH is too small to transfer all of the page, the page length shall not be adjusted to reflect the truncation.

VERSION NUMBER field

The VERSION NUMBER is a one-byte short form notation for the 24-byte assignment in the Firmware Numbers page. Version Numbers are registered by Engineering services.

BEHAVIOR CODE and BEHAVIOR CODE VERSION NUMBER fields

The BEHAVIOR CODE and BEHAVIOR CODE VERSION NUMBER are jointly assigned by the Firmware Engineering Managers of all SCSI design locations.

ASCII FAMILY NUMBER field

The ASCII FAMILY NUMBER field contains the drive model number. The data is left-aligned within this field.

NUMBER OF INTERLEAVES field

The NUMBER OF INTERLEAVES field indicates the number of data interleaves used by the ECC correction algorithm.

DEFAULT NUMBER OF CACHE SEGMENTS field

The DEFAULT NUMBER OF CACHE SEGMENTS field gives the number of segments into which the host requests the drive divide the cache.

4.4.6 Extended INQUIRY Data VPD page (86h)

The Extended INQUIRY Data VPD page (see table 333) provides the application client with a means to obtain information about the logical unit.

Table 333. Extended INQUIRY Data VPD page

Byte \ Bit	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (86h)							
2	Reserved							
3	PAGE LENGTH (3Ch)							
4	Reserved		SPT			GRD_CHK	APP_CHK	REF_CHK
5	Reserved			GROUP_SUP	PRIOR_SUP	HEADSUP	ORDSUP	SIMPSUP
6	Reserved						NV_SUP	V_SUP
7 … 63	Reserved							

PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 333.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data and shall be set to 60. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

SPT (Supported Protection Type) field

A supported protection type (SPT) field (see table 334) indicates the type of protection the logical unit supports. The SPT field shall be ignored if the PROTECT bit is set to zero.

Table 334. SPT field

Code	Definition
000b	The logical unit supports type 1 protection (see SBC-3).
001b	The logical unit supports type 2 protection (see SBC-3).
010b	Reserved
011b	The logical unit supports type 3 protection (see SBC-3).
100b - 111b	Reserved

GRD_CHK (Guard Check) bit

0 A guard check (GRD_CHK) bit set to zero indicates that the device server does not check the LOGICAL BLOCK GUARD field in the protection information (see SBC-3), if any.

1 A GRD_CHK bit set to one indicates that the device server checks the LOGICAL BLOCK GUARD field in the protection information, if any.

APP_CHK (Application Tag Check) bit

0 An application tag check (APP_CHK) bit set to zero indicates that the device server does not check the LOGICAL BLOCK APPLICATION TAG field in the protection information (see SBC-3), if any.

1 An APP_CHK bit set to one indicates that the device server checks the LOGICAL BLOCK APPLICATION TAG field in the protection information, if any.

REF_CHK (Reference Tag Check) bit

0 A reference tag check (REF_CHK) bit set to zero indicates that the device server does not check the LOGICAL BLOCK REFERENCE TAG field in the protection information (see SBC-3), if any.

1 A REF_CHK bit set to one indicates that the device server checks the LOGICAL BLOCK REFERENCE TAG field in the protection information, if any.

GROUP_SUP (Grouping Function Supported) bit

0 A grouping function supported (GROUP_SUP) bit set to one indicates that the grouping function (see SBC-3) is supported by the device server.

1 A GROUP_SUP bit set to zero indicates that the grouping function is not supported.

PRIOR_SUP (Priority Supported) bit

1 A priority supported (PRIOR_SUP) bit set to one indicates that task priority (see SAM-4) is supported by the logical unit.

0 A PRIOR_SUP bit set to zero indicates that task priority is not supported.

HEADSUP (Head of Queue Supported) bit

1 A head of queue supported (HEADSUP) bit set to one indicates that the HEAD OF QUEUE task attribute (see SAM-4) is supported by the logical unit.

0 A HEADSUP bit set to zero indicates that the HEAD OF QUEUE task attribute is not supported. If the HEADSUP bit is set to zero, application clients should not specify the HEAD OF QUEUE task attribute as an Execute Command procedure call argument.

ORDWUP (Ordered Supported) bit

1 An ordered supported (ORDSUP) bit set to one indicates that the ORDERED task attribute (see SAM-4) is supported by the logical unit.

0 An ORDSUP bit set to zero indicates that the ORDERED task attribute is not supported. If the ORDSUP bit is set to zero, application clients should not specify the ORDERED task attribute as an Execute Command procedure call argument.

SIMPSUP (Simple Supported) bit

1 A simple supported (SIMPSUP) bit set to one indicates that the SIMPLE task attribute (see SAM-4) is supported by the logical unit. Logical units that support the full task management model (see SAM-4) shall set the SIMPSUP bit to one.

0 A SIMPSUP bit set to zero indicates that the SIMPLE task attribute is not supported. If the SIMPSUP bit is set to zero, application clients should not specify the SIMPLE task attribute as an Execute Command procedure call argument.

Note. SAM-4 defines how unsupported task attributes are processed.

NV_SUP bit

1 An NV_SUP bit set to one indicates that the device server supports a non-volatile cache and that the applicable command standard defines features using this cache (e.g., the FUA_NV bit in SBC-3).

0 An NV_SUP bit set to zero indicates that the device server may or may not support a non-volatile cache.

V_SUP bit

1 A V_SUP bit set to one indicates that the device server supports a volatile cache and that the applicable command standard defines features using this cache (e.g., the FUA bit in SBC-3).

0 An V_SUP bit set to zero indicates that the device server may or may not support a volatile cache.

4.4.7 Firmware Numbers page (C0h)

Table 335 applies to drives that do not supply a servo RAM Release number and date and a ROM Release date. Table 336 applies to model families other than those covered by Table 335.

Table 335. Firmware Numbers page (C0h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER		PERIPHERAL DEVICE TYPE					
1	PAGE CODE (C0h)							
2	00h							
3	PAGE LENGTH (18h)							
4 : 11	SCSI FIRMWARE RELEASE NUMBER							
12 : 19	ASCII space characters							
20 : 27	Servo ROM Release Number							

PERIPHERAL QUALIFIER field and PERIPHERAL DEVICE TYPE fields

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 335.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

SCSI FIRMWARE RELEASE NUMBER field

The SCSI FIRMWARE RELEASE NUMBER field contains ASCII data. The least significant ASCII character of the drive firmware number shall appear as the last byte of a successful data transfer.

Note. The above information is for drives that return only 28 bytes. For drives that return more than 28 bytes, see Table 336.

Table 336. Firmware Numbers page (C0h) (Applies to model families not covered by Table 335)

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER		PERIPHERAL DEVICE TYPE					
1	PAGE CODE (C0h)							
2	00h							
3	PAGE LENGTH							
4 : 11	SCSI FIRMWARE RELEASE NUMBER							
12 : 19	SERVO FIRMWARE RELEASE NUMBER							
20 : 27	SAP BLOCK POINT NUMBERS (MAJOR/MINOR)							
28 : 31	SERVO FIRMWARE RELEASE DATE							
32 : 35	SERVO ROM RELEASE DATE							
36 : 43	SAP FIRMWARE RELEASE NUMBER							

Bit Byte	7	6	5	4	3	2	1	0
44 : 47	SAP FIRMWARE RELEASE DATE							
48 : 51	SAP FIRMWARE RELEASE YEAR							
52 : 55	SAP MANUFACTURING KEY							
56 : 59	SERVO FIRMWARE PRODUCT FAMILY AND PRODUCT FAMILY MEMBER IDs							

PERIPHERAL QUALIFIER field and PERIPHERAL DEVICE TYPE fields

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 336.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

SCSI FIRMWARE RELEASE NUMBER fields

The SCSI FIRMWARE RELEASE NUMBER fields contain ASCII data. The least significant ASCII character of the Drive firmware number shall appear as the last byte of a successful data transfer.

SERVO ROM RELEASE DATE fields

Applies to model families that return 36 bytes rather than 28. Newer drive models return 60 bytes. Servo firmware RAM and ROM Release dates are added in some models. See “Vital Product Data pages supported” table in the Product Manual, Volume 1, that applies to the drive in question.

SERVO ADAPTIVE PARAMETERS (SAP) fields

Servo Adaptive Parameters (SAP) are created independently from the servo firmware. Thus, they have their own release information. The SAP Firmware Release Number field contains ASCII data.

SAP MANUFACTURING KEY field

The SAP MANUFACTURING KEY field contains a binary code used by manufacturing to identify the SAP.

SERVO FIRMWARE PRODUCT FAMILY and PRODUCT FAMILY MEMBER IDs field

The high order word of the SERVO FIRMWARE PRODUCT FAMILY and PRODUCT FAMILY MEMBER IDs field contains a binary code which corresponds to the Servo Firmware Family ID. The low order word of this field contains a binary code which corresponds to the Product Family Member ID.

Note. Applies to model families that return 59 bytes rather than 35 or 28. Servo firmware RAM and ROM Release dates are added, plus SAP firmware. See Vital Product Data pages supported” table in the Product Manual, Volume 1, that applies to the drive in question.

4.4.8 Device Identification VPD page (83h)

The Device Identification VPD page (see table 337) provides the means to retrieve zero or more identification descriptors applying to the logical unit. Logical units may have more than one identification descriptor (e.g., if several types or associations of identifier are supported). Device identifiers consist of one or more of the following:

- Logical unit names;
- SCSI target port identifiers;
- SCSI target port names;
- SCSI target device names;
- Relative target port identifiers;
- SCSI target port group number; or
- Logical unit group number.

Identification descriptors shall be assigned to the peripheral device (e.g., a disc drive) and not to the currently mounted media, in the case of removable media devices. Operating systems are expected to use the identification descriptors during system configuration activities to determine whether alternate paths exist for the same peripheral device.

Table 337. Device Identification VPD page

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (83h)							
2	MSB	PAGE LENGTH (n-3)						
3								LSB
	DENTIFICATION DESCRIPTOR LIST							
4	IDENTIFICATION DESCRIPTOR (First)							
:	:							
:	:							
:	:							
n	IDENTIFICATION DESCRIPTOR (Last)							

PERIPHERAL QUALIFIER and the PERIPHERAL DEVICE fields

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 337.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

IDENTIFICATION DESCRIPTOR LIST field

The IDENTIFICATION DESCRIPTOR LIST provides a list of Identification Descriptor information, the format of which is given in Table 338.

IDENTIFICATION DESCRIPTOR field

Each IDENTIFICATION DESCRIPTOR (see Table 338) contains information identifying the logical unit, physical device, or access path used by the command and returned parameter data.

Table 338. Identification Descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	PROTOCOL IDENTIFIER				CODE SET			
1	PIV	Reserved	ASSOCIATION		IDENTIFIER TYPE			
2	Reserved							
3	IDENTIFIER LENGTH (n – 3)							
4	IDENTIFIER							
n								

PROTOCOL IDENTIFIER field

The PROTOCOL IDENTIFIER field may indicate the SCSI transport protocol to which identification descriptor applies. If the ASSOCIATION field contains a value other than 1h or 2h or the PIV bit is set to zero, then the PROTOCOL IDENTIFIER field should be ignored. If the ASSOCIATION field contains a value of 1h or 2h and the PIV bit is set to one, then the PROTOCOL IDENTIFIER field shall contain one of the values shown in table 238 (see 7.5.1) to indicate the SCSI transport protocol to which identification descriptor applies.

CODE SET field

The CODE SET field specifies the code set used for the identifier field, as described in Table 339. This field is intended to be an aid to software that displays the identifier field.

Table 339. Code Set

Value	Description
0h	Reserved
1h	The IDENTIFIER field shall contain binary values
2h	The IDENTIFIER field shall contain ASCII graphic codes (i.e., code values 20h through 7Eh)
3h - Fh	Reserved

PIV (Protocol Identifier Valid) bit

0 A protocol identifier valid (PIV) bit of zero indicates the PROTOCOL IDENTIFIER field should be ignored. If the ASSOCIATION field contains a value of 1h or 2h then a PIV bit set to one indicates the PROTOCOL IDENTIFIER field contains a valid protocol identifier selected from the values shown in table 340. If the ASSOCIATION field contains a value other than 1h or 2h then the PIV bit should be ignored.

Table 340. PROTOCOL IDENTIFIER values

Protocol Identifier	Description	Protocol Standard
0h	Fibre Channel	FCP-2
1h	Parallel SCSI	SPI-4
2h	SSA	SSA-S3P
3h	IEEE 1394	SBP-2
4h	Remote Direct Memory Access (RDMA)	SRP
5h	Internet SCSI	iSCSI
6h	SAS Serial SCSI Protocol	SAS
7h - Fh	Reserved	

ASSOCIATION field

The ASSOCIATION field indicates the entity that the Identification descriptor describes. If a physical or logical device returns an Identification descriptor with the ASSOCIATION field set to 0h, it shall return the same descriptor when it is accessed through any other path.

The ASSOCIATION field specifies the entity with which the Identifier field is associated, as described in Table 341.

Table 341. ASSOCIATION field

Value	Description
0h	The IDENTIFIER field is associated with the addressed physical or logical device
1h	The IDENTIFIER field is associated with the port that received the request
2h	The IDENTIFIER field is associated with the SCSI target device that contains the addressed logical unit.
‘3h	Reserved

IDENTIFIER TYPE field

The IDENTIFIER TYPE field specifies the format and assignment authority for the identifier, as described in tables 342 and 343.

Table 342. IDENTIFIER Type field

Value	Description
0h	No assignment authority was used and consequently there is no guarantee that the identifier is globally unique (i.e., the identifier is vendor specific).
1h	The first 8 bytes of the IDENTIFIER field are a Vendor ID (see annex C). The organization associated with the Vendor ID is responsible for ensuring that the remainder of the identifier field is unique. One recommended method of constructing the remainder of the identifier field is to concatenate the product identification field from the standard INQUIRY data field and the product serial number field from the unit serial number page.
2h	The IDENTIFIER field contains a Canonical form IEEE Extended Unique Identifier, 64-bit (EUI-64). In this case, the identifier length field shall be set to 8. Note that the IEEE guide-lines for EUI-64 specify a method for unambiguously encapsulating an IEEE 48-bit identifier within an EUI-64.
3h	The IDENTIFIER field contains an FC-PH, FC-PH3 or FC-FS Name_Identifier. Any FC-PH, FC-PH3 or FC-FS identifier may be used, including one of the four based on a Canonical form IEEE company_id.
4h	If the ASSOCIATION field contains 1h, the Identifier value contains a four-byte binary number identifying the port relative to other ports in the device using the values shown Table 341. The CODE SET field shall be set to 1h and the IDENTIFIER LENGTH field shall be set to 4h. If the ASSOCIATION field does not contain 1h, use of this identifier type is reserved.
5h	If the Association value is 1h, the Identifier value contains a four-byte binary number identifying the port relative to other ports in the device using the values shown Table 341. The CODE SET field shall be set to 1h and the IDENTIFIER LENGTH field shall be set to 4h. If the ASSOCIATION field does not contain 1h, use of this identifier type is reserved.
6h	If the ASSOCIATION value is 0h, the IDENTIFIER value contains a four-byte binary number identifying the port relative to other ports in the device using the values shown Table 341. The CODE SET field shall be set to 1h and the IDENTIFIER LENGTH field shall be set to 4h. If the ASSOCIATION field does not contain 0h, use of this identifier type is reserved.
7h	The MD5 logical unit identifier shall not be used if a logical unit provides unique identification using identifier types 2h or 3h. A bridge device may return a MD5 logical unit identifier type for that logical unit that does not support the Device Identification VPD page.
8h - Fh	Reserved.

IDENTIFIER LENGTH field

The IDENTIFIER LENGTH field specifies the length in bytes of the IDENTIFIER field. If the ALLOCATION LENGTH field of the command descriptor block is too small to transfer all of the identifier, the identifier length shall not be adjusted to reflect the truncation.

IDENTIFIER field

The IDENTIFIER field contains the identifier as described by the Association, Identifier Type, CODE SET, and IDENTIFIER LENGTH fields. The example described in this clause and shown in Table 343 is not a normative part of this manual. This example of a complete device identification VPD page assumes that the product is a direct-access device with an T10 Vendor ID of “XYZ_Corp,” a product identification of “Super Turbo Disk,” and a product serial number of “2034589345.” Furthermore, it is assumed that the manufacturer has been assigned a 24-bit IEEE company_id of 01ABCDh by the IEEE Registration Authority Committee and that the manufacture has assigned a 24-bit extension_identifier of 234567h to this logical unit. The combined 48-bit identifier is reported in the 64-bit format as defined by the IEEE 64-bit Global Identifier (EUI-64) standard. The data returned in the device identification VPD page for this logical unit is shown in Table 343.

Table 343. Device Identification page example

Bytes	Hexadecimal values																ASCII values
00–15	00	83	00	32	02	01	00	22	58	59	5A	5F	43	6F	72	70	...2...XYZ_Corp [5]
16–31	53	75	70	65	72	20	54	75	72	62	6F	20	44	69	73	6B	Super Turbo Disk
32–47	32	30	33	34	35	38	39	33	34	35	01	02	00	08	01	AB	2034589345.......
48–53	CD	FF	FF	23	45	67											......

Notes.

a) Non-printing ASCII characters are shown as ".".

b) Byte 00 is the beginning of the VPD page (see Table 328).

c) Byte 04 is the beginning of the Identification descriptor for the Vendor ID based identifier (Identifier type 1 see Table 342).

d) Byte 42 is the beginning of the Identification Descriptor for the EUI-64 identifier (Identifier type 2, see Table 342).

e) For Seagate devices, this will say "Seagate."

4.4.9 Jumper Settings page (C2h)

Table 344. Jumper Settings page (C2h)

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (C2h)							
2	00h							
3	PAGE LENGTH (02h)							
4	DS	MS	WP	PE	DRIVE ID			
5	Reserved							TE

PERIPHERAL QUALIFIER and PERIPHERAL DEVICE TYPE fields

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 344.

PAGE LENGTH field

The PAGE LENGTH field specifies the length of the following VPD page data. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

DS (Delayed Motor Start) bit

The Delayed Motor Start (DS) bit when set to 1 indicates that this jumper is on.

MS (Motor Start) bit

The Motor Start (MS) bit when set to 1 indicates that the jumper is on.

WP (Write Protect) bit

The Write Protect (WP) bit when set to 1 indicates that the write protect jumper is on.

PE (Parity Enable) and PD (Parity Disable) bits

On some drives, Parity Enable (PE) bit when set to 1 indicates that SCSI parity error checking jumper is on. Other drives have a PD (PARITY DISABLE) jumper, where Parity is enabled when the jumper is off. See individual drive's Product Manual, Volume 1, or Installation Guide for information on how the drive of interest is set up.

Drive ID field

The Drive ID is shown below in Table 345. Bit 3 is the most significant bit and bit 0 is the least significant bit.

Table 345. Drive ID bit

Bit 3	Bit 2	Bit 1	Bit 0	Drive ID
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

TE (Terminator Enable) bit

Terminator Enable (TE) is not used on all drives. See individual drive’s Product Manual, Volume 1, for applicability.

Note. It is not presently possible to return information on the Terminator Power (TP) jumpers.

4.4.10 Mode Page Policy VPD page

The Mode Page Policy VPD page (see table 346) indicates which mode page policy (see table 3.11) is in effect for each mode page supported by the logical unit.

Table 346. Mode Page Policy VPD page

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (87h)							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
	Mode page policy descriptor list							
4	Mode page policy descriptor [first]							
7								
	.. .							
n-3	Mode page policy descriptor [last]							
n								

PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 346.

PAGE LENGTH field

The PAGE LENGTH field indicates the length of the mode page policy descriptor list. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

Each mode page policy descriptor (see table 347) contains information describing the mode page policy for one or more mode pages or subpages (see 4.3.5). The information in the mode page policy descriptors in this VPD page shall describe the mode page policy for every mode page and subpage supported by the logical unit.

Table 347. Mode page policy descriptor

Bit / Byte	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							

POLICY PAGE CODE field and POLICY SUBPAGE CODE field

The POLICY PAGE CODE field and POLICY SUBPAGE CODE field indicate the mode page and subpage to which the descriptor applies.

If the first mode page policy descriptor in the list contains a POLICY PAGE CODE field set to 3Fh and a POLICY SUBPAGE CODE field set to FFh, then the descriptor applies to all mode pages and subpages not described by other mode page policy descriptors. The POLICY PAGE CODE field shall be set to 3Fh and the POLICY SUBPAGE CODE field shall be set to FFh only in the first mode page policy descriptor in the list.

If the POLICY PAGE CODE field contains a value other than 3Fh and a POLICY SUBPAGE CODE field contains a value other than FFh, then the POLICY PAGE CODE field and the POLICY SUBPAGE CODE field indicate a single mode page and subpage to which the descriptor applies.

If the POLICY PAGE CODE field contains a value other than 3Fh, then POLICY SUBPAGE CODE field shall contain a value other than FFh. If the POLICY SUBPAGE CODE field contains a value other than FFh, then POLICY PAGE CODE field shall contain a value other than 3Fh.

MLUS bit

If the SCSI target device has more than one logical unit, a multiple logical units share (MLUS) bit set to one indicates the mode page and subpage identified by the POLICY PAGE CODE field and POLICY SUBPAGE CODE field is shared by more than one logical unit. A MLUS bit set to zero indicates the logical unit maintains its own copy

The MLUS bit is set to one in the mode page policy descriptors or descriptor that indicates the mode page policy for the:

a) Disconnect-Reconnect mode page (see 4.3.10); and
b) Protocol Specific Port mode page (see 4.3.17).

MODE PAGE POLICY field

The MODE PAGE POLICY field (see table 348) indicates the mode page policy for the mode page and subpage identified by the POLICY PAGE CODE field and POLICY SUBPAGE CODE field. The mode page policies are described in table 63 (see 3.11).

Table 348. MODE PAGE POLICY field

Code	Description
00b	Shared
01b	Per target port
10b	Obsolete
11b	Per I_T nexus

4.4.11 Supported Vital Product Data pages (00h)

This clause contains a list of the vital product data page codes supported by the target or logical unit (see Table 349). If a target supports any vital product data pages, it also shall support this vital product data page.

Table 349. Supported Vital Product Data pages

Bit / Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (00h)							
2	Reserved							
3	PAGE LENGTH (n-3)							
4	SUPPORTED PAGE LIST							
n								

PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

The PAGE CODE field shall be set to the value defined in table 349.

PAGE LENGTH field

The PAGE LENGTH field indicates the length of the mode page policy descriptor list. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 2.2.6.

SUPPORTED PAGE LIST field

The SUPPORTED PAGE LIST field shall contain a list of all vital product data page codes (see clause 4.4.1) implemented for the target or logical unit in ascending order beginning with page code 00h.

4.4.12 Unit Serial Number page (80h)

This page provides a product serial number for the target or logical unit. See Table 350 following.

Table 350. Unit Serial Number page (80h)

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (80h)							
2	Reserved							
3	PAGE LENGTH							
4 - n	Product Serial Number							

PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are defined in STANDARD INQUIRY DATA (see 3.6.2).

PAGE CODE field

PAGE CODE field

PAGE LENGTH field

The PAGE LENGTH field specifies the length in bytes of the product serial number page. Older products that only support the Product Serial Number parameter will have a page length of 08h, while newer products that support both parameters will have a page length of 14h. If the ALLOCATION LENGTH is too small to transfer all of the page, the page length shall not be adjusted to reflect the truncation.

Product Serial Number field

The Product Serial Number field contains ASCII data that is vendor-assigned serial number. The least significant ASCII character of the serial number shall appear as the last byte in the Data-In Buffer. If the product serial number is not available, the target shall return ASCII spaces (20h) in this field.

Index

D

E

F

R

Seagate Technology LLC
920 Disc Drive, Scotts Valley, California 95066-4544, USA
Publication Number: 100293068, Rev. C