American National Standard

for information systems –

intelligent peripheral interface – device-generic command set for magnetic and optical disk drives

Adopted for Use by the Federal Government

FIPS PUB 130

See Notice on Inside Front Cover

american national standards institute, inc.
1430 broadway, new york, new york 10018

ANSI X3.132-1987

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 130, Intelligent Peripheral Interface (IPI). For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator (ADP), Institute for Computer Sciences and Technology, National Bureau of Standards, Gaithersburg, MD 20899.

ANSI®
X3.132-1987

American National Standard
for Information Systems –

Intelligent Peripheral Interface – Device-Generic Command Set for Magnetic and Optical Disk Drives

Secretariat

Computer and Business Equipment Manufacturers Association

Approved December 15, 1986

American National Standards Institute, Inc

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

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

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

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

American National Standards Institute
1430 Broadway, New York, New York 10018

Printed in the United States of America

PC1M588/25

Foreword

(This Foreword is not part of American National Standard X3.132-1987.)

This standard provides a definition of the device-generic portion of a family of standards called the Intelligent Peripheral Interface (IPI). It is a new high-performance, general-purpose parallel peripheral interface. This standard responds to an industry market need (expressed both by users and manufacturers) to limit the increasing costs in hosts associated with changes in peripherals. The intent of the IPI is to isolate the host (CPU), both hardware and software, from changes in peripherals by providing a "function-generic" command set to allow the connection of multiple types of peripherals (disks, printers, tapes, communications). To smooth the transition from the current methods to the generic approach, the IPI also supports device-specific command sets to aid in bridging the gap between the two approaches.

To accomplish this set of goals, the design of the IPI includes device-specific and device-generic command sets, both utilizing a common physical bus. The device-specific command set provides:

(1) Device-oriented control
(2) Physical Data Addressing
(3) Timing Critical Operations
(4) Lower Device Cost

The device-generic command set provides a higher level of functionality and portability. It includes:

(1) Host/Device Independence
(2) Logical Data Addressing
(3) Timing Independence
(4) Command Queuing Capability

A system is not restricted to the use of one level of command set or the other. It is possible that both levels of command sets will be utilized with a given system's architecture to balance such parameters as system performance, cost, and peripheral availability. It is also possible for the host to provide for migration from device-specific to device-generic levels while still retaining the same physical interface.

The development of an Intelligent Peripheral Interface (IPI) was begun after a preliminary investigation had been completed. The earliest proposals were made by participants of Task Group X3T9.3 in late 1978. At that time, the Task Group decided generic-oriented peripheral interfaces were not yet ready for standardization and that the group should concentrate on device-oriented interfaces and the system-oriented, high-speed serial interfaces. The group acknowledged the desirability of higher level intelligent commands by reserving code fields in American National Standard for Information Systems — Interface between Rigid Disk Drive(s) and Host(s), ANSI X3.101-1984, during its April 1980 meeting.

The basic architecture of the resultant IPI was first proposed at the X3T9.3 August 1980 meeting. In addition to the 1978 proposal, complete company implementations were proposed by several manufacturers from August 1980 to August 1981. These proposals resulted from the initiative of the contributors and from wide-spread solicitation by the task group.

Task Group X3T9.3 agreed upon preliminary functional requirements during the October 1980 meeting, which included the following:

(1) Parallel transfer
(2) Command and Data Handshaking
(3) Allowance for high-speed transfers without Handshaking
(4) Transfer rate up to 10 Megaoctets per second

Task Group X3T9.3 began work on the IPI in 1981 in response to an emerging need for a higher performance peripheral interface. Coincidental with the need for higher performance was the availability of low-cost VLSI circuit technologies, allowing increased intelligence in the peripheral device. These needs were confirmed by large and active participation from all areas of the computer industry.

The fundamental characteristics that the group achieved included the following:

(1) Single or dual octet transfers
(2) Data rates of at least 10 megabytes per second
(3) Cable lengths extending from 5 to 125 meters depending upon type of transmitter and cable
(4) Low-cost, commonly available components
(5) High level of maintainability and availability
(6) A multilevel command structure allowing different levels of intelligence in the peripherals
(7) A definition that facilitates evolutionary changes in the levels with minimal impact on software and hardware components
(8) Definitions supporting an extensive group of peripheral devices including disks, tape, communications equipment, printers, and the like, with a common choice of interface hardware and commands

Suggestions for improvement of this standard will be welcome. They should be sent to the Computer and Business Equipment Manufacturers Association, 311 First Street, NW, Suite 500, Washington, D.C. 20001.

This standard was processed and approved for submittal to ANSI by Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not imply that all committee members voted for approval. At the time it approved this standard, the X3 Committee had the following members:

Edward Lohse, Chair
Richard Gibson, Vice-Chair
Catherine A. Kachurik, Administrative Secretary

Organization Represented	*Name of Representative*
American Express	D. L. Seigal Lucille Durfee (Alt)
American Library Association	Paul Peters
American Nuclear Society	Geraldine C. Main D. R. Vondy (Alt)
AMP Incorporated	Edward Kelly Patrick E. Lannan (Alt)
Association for Computing Machinery	Kenneth Magel Jon A. Meads (Alt)
Association of the Institute for Certification of Computer Professionals	
AT&T Communications	Thomas M. Kurihara Henry L. Marchese Richard Gibson (Alt)

Organization Represented	*Name of Representative*
AT&T Information Systems	Herbert V. Bertine
	Paul D. Bartoli (Alt)
	Stuart M. Garland (Alt)
Burroughs Corporation	Stanley Fenner
Control Data Corporation	Charles E. Cooper
	Keith Lucke (Alt)
Cooperating Users of Burroughs Equipment	Thomas Easterday
	Donald Miller (Alt)
Data General Corporation	Michael W. Kelley
	Lyman Chapin (Alt)
Data Processing Management Association	Ward Arrington
	Wallace W. McPherson (Alt)
Digital Equipment Computer Users Society	William Hancock
	Dennis Perry (Alt)
Digital Equipment Corporation	Gary S. Robinson
	Delbert L. Shoemaker (Alt)
Eastman Kodak	Gary Haines
	Charleton C. Bard (Alt)
General Electric Company	Richard W. Signor
	William R. Kruesi (Alt)
General Services Administration	William C. Rinehuls
	Larry L. Jackson (Alt)
GUIDE International	Frank Kirshenbaum
	Sandra Swartz Abraham (Alt)
Harris Corporation	Walter G. Fredrickson
	Rajiv Sinha (Alt)
Hewlett-Packard	Donald C. Loughry
Honeywell Information Systems	Thomas J. McNamara
	David M. Taylor (Alt)
IBM Corporation	Mary Anne Gray
	Robert H. Follett (Alt)
IEEE Computer Society	Sava I. Sherr
	Thomas M. Kurihara (Alt)
	Thomas A. Varetoni (Alt)
Lawrence Berkeley Laboratory	David F. Stevens
	Robert L. Fink (Alt)
Moore Business Forms	Delmer H. Oddy
National Bureau of Standards	Robert E. Rountree
	James H. Burrows (Alt)
National Communications System	George W. White
NCR Corporation	A. Raymond Daniels
	Thomas W. Kern (Alt)
Prime Computer, Inc	Arthur Norton
	Linda J. Allen (Alt)
Railinc Corporation	R. A. Petrash
Recognition Technology Users Association	Herbert F. Schantz
	G. W. Wetzel (Alt)
Scientific Computer Systems Corporation	James A Baker
	Carl Haberland (Alt)
SHARE, Inc	Thomas B. Steel
	Robert A. Rannie (Alt)
Sperry Corporation	Marvin W. Bass
	Jean G. Smith (Alt)
Texas Instruments, Inc	Presley Smith
	Richard F. Trow, Jr (Alt)
3M Company	Paul D. Jahnke
	J. Wade Van Valkenburg (Alt)
Travelers Insurance Companies, Inc	Joseph T. Brophy
U.S. Department of Defense	Fred Virtue
	Belkis Leong-Hong (Alt)
VIM	Chris Tanner
	Madeleine Sparks (Alt)
VISA U.S.A.	Jean T. McKenna
	Susan Crawford (Alt)
Wang Laboratories, Inc	Marsha Hayek
	Joseph St. Amand (Alt)
Xerox Corporation	John L. Wheeler
	Roy Pierce (Alt)

Subcommittee X3T9 on I/O interfaces, which reviewed this standard, had the following members:

Delbert L. Shoemaker, Chair
William E. Burr, Vice-Chair

Dennis Appleyard
James R. Barnette
Duane Barney
Steve Cooper
Louis C. Domshy
Robert Dugan
Reinhard Knerr
Patrick Lannan
John B. Lohmeyer
John McCool
Gene Milligan
Ted Petrowich
Gary S. Robinson
Arnold J. Roccati
Floyd E. Ross
Robert B. Anthony (Alt)
Charles Brill (Alt)
George Clark (Alt)
Roger Cormier (Alt)
Mark Hammang (Alt)
John Hancock (Alt)
Sunil Joshi (Alt)
Kris Kowal (Alt)
Dennis Krob (Alt)
Kirk Moulton (Alt)

Task Group X3T9.3 on Device Level Interfaces, which was responsible for the development of this standard, had the following participants:

Gary S. Robinson, Chair
I. Dal Allan, Vice-Chair

J. Amstutz
D. Appleyard
R. Barnes
D. Barney
R. Bender
R. Bergey
F. Berkowitz
B. Bonner
M. Bradac
C. Brill
B. Brown
R. Brown
W. Burr
E. Calkins
C. Chen
E. Cieniawa
S. Cooper
R. Davideit
R. Davis
R. Derr
S. Dick
R. Dillon
R. Driscal
T. Eiland
D. Filpus
S. Finch
R. Fish
M. Fitzpatrick
M. Gamerl
R. Geller
S. Gersten
M. Glier
W. Grace
B. Graham
E. Grivna
D. Guss
K. Hallam
M. Hammang
D. Hartig
P. Hayden
C. Hess
C. Jarboe
S. Juhasz
D. Klang
K. Kong
A. Kononov
T. Leland
J. Lohmeyer
R. Lopez
J. Luttrull
R. Matheson
T. McClendon
D. McIntyre
P. Mclean
F. Meadows
J. Meyer
G. Milligan
P. Mizera
D. Moczarny
K. Moe
J. Monaco
R. Morris
J. Mulligan
R. Notari
T. O'Connor
M. O'Donnell
J. Patton
R. Peacock
J. Peterson
T. Petrowich
P. Phillips
M. Poehler
D. Ray
B. Reago
C. Ridgeway
D. Roberts
W. Roberts
A. Roccati
F. Ross
L. Russell
A. Salthouse
W. Sanderson
E. Sandoval
K. Scharf
D. Schneider
J. Schuessler
R. Schultz
D. Shoemaker
E. Slater
J. Smith
R. Snively
C. Stead
H. Stehle
M. Stewart
H. Truestedt
D. Tsai
N. Vashi
D. Voigt
C. Walker
O. Weeden
D. Williams
L. Zorza

Contents

SECTION PAGE

American National Standard
for Information Systems –

Intelligent Peripheral Interface – Device-Generic Command Set for Magnetic and Optical Disk Drives

1. Scope and Editorial Conventions

1.1 Scope. This document describes the Logical Level 3 (device-level) Interface for disk drives. The physical, electrical, and configuration characteristics and the transmission protocol of this interface are in accordance with American National Standard for Information Systems – Intelligent Peripheral Interface – Physical Level, ANSI X3.129-1986. The interface is capable of handling data rates from 0 to at least 10 megaoctets per second, depending on driver and receiver classes.

The purpose of this standard is to facilitate the development and utilization of a device-level interface that permits the interconnection of disk slave peripherals to a controller.

1.2 Editorial Conventions. Certain terms used in this standard that are proper names of signals are printed in uppercase to avoid possible confusion with other uses of the same words; e.g., ATTENTION IN. Any lowercase uses of these words have the normal English meaning.

A number of conditions, sequence parameters, events, English text, states, or similar terms are printed with the first letter of each word in uppercase and the rest lowercase; e.g., In, Out, Selective Reset, Bidirectional, Bus Control, Operation Response. Any lowercase uses of these words have the normal English meaning.

1.3 Description of Sections. The first five sections contain material that is useful across all classes of device that the Device-Generic command sets can support. The later sections are oriented to particular device classes. This document is intended for use with Magnetic and Optical disks.

Section 4 describes the Environment of Use and projected application areas.

Section 5 describes the Message Packet structure used for commands and responses.

Section 6 describes Control commands.

Section 7 describes Position commands.

Section 8 describes the most generic Transfer commands.

Section 9 describes the Combination Transfer commands, which require a minimum of two sets of extents.

Section 10 describes the other Transfer commands, which are more device specific than those in Section 6.

Section 11 describes the Diagnostic commands.

Section 12 summarizes the commands defined in the document.

Appendixes contain information that is not part of the standard, but complementary to, and considered useful material for the reader.

1.4 Interface Levels. The IPI has adopted a layered approach to functionality, described as levels. In Section 6 of American National Standard for Information Systems - Intelligent Peripheral Interface - Physical Level, ANSI X3.129-1986, the relationship between the various levels is described. The description of Level 3 (6.2.2 in ANSI X3.129-1986) is as follows:

6.2.2 Level 3 - Device Generic. Level 3 is oriented to the generic components of devices (disk, tape, printer, and the like), and typically not the device-unique components (e.g., cylinders, heads). Level 3 uses a packet structure that provides independence of the command repertoire from the Physical Interface. Some of the Level 3 characteristics are:

(1) Operations may be Individual or Queued.

(2) The data area is defined by the facility

addressibility to the media and is specified by the attributes of the slave.

(3) The Transfer of data may or may not be timing critical, and typically is buffered.

(4) Data is addressed by DataBlock address, but PhysicalBlock addresses may be used.

(5) DataBlock lengths typically are fixed over an addressable data area, and can vary between different addressable areas on the media as well as varying between medias. Tape DataBlocks may vary between any two blocks.

(6) Data is normally requested as "perfect" (data errors, if any, corrected), or may be requested as "raw" (data errors, if any, are not corrected).

(7) Positioning is requested implicitly, but may be explicit.

(8) Media defect handling is transparent to the master, but may be managed by the master.

(9) Error correction is transparent to the master, but may be managed by the master.

(10) Error retry is transparent to the master, but may be managed by the master.

1.5 Conceptual Overview. These concepts are described relative to their usage in a Device-Generic environment. These relationships may be defined differently in other levels of IPI implementation.

1.5.1 Relationship of Master, Slave, and Facility. A master is an entity that has need for some form of information transfer or storage. The master makes use of slaves and facilities to perform the needed operation. The slave or facility (addressee) to which the master addresses a service request is expected to have sufficient intelligence and capability to perform the requested service.

A slave is subservient to one master per port. The slave provides services to the attached master or masters, and is responsible for the control and operation of the facilities (if any) attached to it. The slave may also perform additional functions as it finds necessary.

A facility may be a unit of storage (e.g., disk drives and tape drives) or a unit of functionality (e.g., a communications interface). Facilities are not limited to these devices, or even to being devices. Facilities are defined as having capabilities. A facility may be subservient to one or more slaves.

1.5.2 Relationship of Facilities and Partitions. As an option, a slave may allow the subdivision of a facility into partitions. In a facility that is capable of being subdivided, there are two possible types of subdivision that could be supported. The first is the ability of the slave to subdivide the facility into a default data partition and up to 7 maintenance partitions.

For disks, the second is the ability of the master to subdivide the default data partition into up to 239 data partitions and up to 8 maintenance partitions.

For tapes, the second is the ability of the slave to define 112 additional partitions and the master to subdivide the default data partition into 127 data partitions and up to 8 maintenanced partitions.

Maintenance partitions are typically used for maintenance purposes, but their use is not restricted thereto, and may be otherwise used for storage of specific data for the slave, the facility, or for both.

Maintenance partitions are not accessed by the master during normal operation, but access by the master is possible via the OPERATING MODE command.

The total area of a facility that is defined by the slave as the default data partition may be used for storage of the master's data. The partitioning and control of the partitions created in the default data area is the responsibility of the master.

1.5.3 Command Structure. The Device-Generic command structure is based on message packets that are of variable length. The basic command and response packets are expanded by appending parameters that identify the specific actions to be taken. This provides a powerful tool for providing flexibility for future growth, since the addition of new functions over time should not require major changes to the command repertoire, but add functionality to the existing commands.

The Device-Specific (known also as Level 2) command repertoire described in American National Standard for Information Systems - Intelligent Peripheral Interface - Device-Specific Command Set for Magnetic Disk Drives, ANSI X3.130-1986, uses a different command structure to the Device-Generic level. The major objective achieved by being differ-ent is the ability to execute timing-critical operations. Because both levels use the same Physical interface, it is possible for Device-Specific slaves to co-exist on the same daisy-chained cable as Device-Generic slaves. If the master daisychains Device-Specific and Device-Generic 3 slaves, this places the requirement on the master to be able to handle both command structures.

Device-Generic commands need to be consistent with the characteristics of the general device type (e.g., the master must know if general device type is a disk or tape). The commands provide for logical data addressing, buffers, error detection, error retry, error correction, and the like, at the slave. There are no timing-critical dependencies between commands.

1.6 Application Environments. The degree of intelligence provided by the slave varies, and is defined by Attributes provided by the manufacturer. The master can either select or respond to the level of intelligence provided by the slave.

It is possible for the manufacturer of a computer system to implement products that are master oriented, and that support software constraints for current operating systems. By adding different degrees of functionality, the manufacturer could migrate over time to a distributed-intelligence, slave-oriented environment.

1.6.1 Control of Facilities by the Master. The master performs Facility Selection and issues Individual commands. This is a master-oriented environment that typifies the operation of most of today's host system I/O interfaces. The master prioritizes tasks and dispenses them to the various peripherals attached to it in a fashion that optimizes the system performance from the master's perspective.

Since the master controls all activity on the interface, it shall poll both the slave and facility interrupts in order to maximize the efficiency of data transfers. The master-oriented environment is typified by the following features of the interface:

(1) Command execution sequenced by the master
(2) Individual commands
(3) Selection to the facility
(4) Master polling of both slaves and facilities
(5) Bus Control established by the master
(6) Command queueing done by the master

1.6.2 Shared Control of Facilities. In this environment, a slave has functional control of the facilities, but the master has explicit control over some aspects of facility management. Some tasks, such as command queuing, may be left to the discretion of the slave, but the master may choose to sequence the flow of data between specific facilities based on its own algorithms.

This environment is typified by the following features of the interface:

(1) Command execution between facilities sequenced by the master
(2) Queued commands
(3) Selection to the facility
(4) Master polling of both slaves and facilities
(5) Bus Control established by the master
(6) Command queuing done by the slave

1.6.3 Control of Facilities by the Slave. In this environment, the slave has functional control of the facilities, and the master has limited control (if any) over facility management. The slave has the intelligence necessary to control operations on behalf of the master.

This is a slave-oriented environment because of the degree of control the slave exercises over system performance and optimization. This environment is typified by the following features of the interface:

(1) Command execution between facilities sequenced by the slave
(2) Queued commands
(3) Selection to the slave
(4) Polling of the slave by the Master
(5) Bus Control established by the master or (optionally) by the slave
(6) Command queuing done by the slave

2. Referenced and Related American National Standards

2.1 Referenced American National Standards. This standard is intended to be used in conjunction with the following American National Standards. When these standards are superseded by a revision approved by the American National Standards Institute, Inc, the revision shall apply:

ANSI X3.129-1986, Information Systems — Intelligent Peripheral Interface — Physical Level

ANSI X3.130-1986, Information Systems — Intelligent Peripheral Interface — Device-Specific Command Set for Magnetic Disk Drives

2.2 Related American National Standards. The following standard is not essential for the completion of the requirements of this standard and are intended to be used solely for explanation or clarification.

ANSI X3.147-1987, Information Systems — Intelligent Peripheral Interface — Device-Generic Command Set for Magnetic Tapes

3. Definitions

actual address. This is the address of the facility, which is unique and is used by the slave to accomplish selection (the interface by which facilities are attached to the slave may or may not be an IPI).

addressee. This term refers to the slave, the facility, or the combination of the two that are identified by the combination of Slave Address and Facility Address.

alias. This term names a partition within a facility.

attributes. Each slave (or facility) has a level of functionality that can be described by a table that contains entries for characteristics that require definition to the master. The entries are self-defining and are termed attributes.

burst. This term describes the maximum number of octets to be transferred in any one Information Transfer. The transfer of a number of bursts may be implicit within a single command request.

command. This term refers to a command issued by the master to initiate some specific operation.

command address. This term refers to the two octets of Slave Address and Facility Address in the basic command packet that uniquely identify (or optionally partition) the addressee to which the command is issued.

data. This term refers to any Information Transfer over the interface not associated with either a command or a response.

DataBlock. This term is uniquely defined in this document as meaning the logical representation of data on the media. A DataBlock may or may not have a relationship to PhysicalBlocks (i.e., it may be equal to, less than, or greater than PhysicalBlocks in size). If commands are issued with DataBlock addresses to a slave that supports logical addressing, the slave is responsible to locate the physical representation of data by logical address correctly, and present it to the master in DataBlocks.

extent. This is a range of contiguous blocks, defined as the count of a number of blocks beginning at a data address (PhysicalBlock or DataBlock) within (and limited to) a partition.

facility address. This is an octet value in the Command Address that may be an Actual Address, Selection Address, Synonym, or Alias. It is used to identify the facility (or partition) to which the command is addressed. Valid values are from 0 to 255.

housekeeping. This is a procedure required at initial usage of a slave (or facility) because the master must obtain the attributes in order to adapt to, or configure, its functionality.

individual. This term refers to the execution of commands that must be completed before the next command can be issued.

information transfer. This term refers to the transferring of octets on the Physical Interface of commands, responses, and data that are framed by a Bus Exchange.

interrupts. This term refers to the ability at the Physical Interface for the slave to advise the master of which types of Response are available in the slave. The slave uses interrupts to initiate assertion of the ATTENTION IN signal.

L-available (logically available). See 4.7.1.7.

L-busy (logically busy). See 4.7.1.9.

level 2 (device specific). This term refers to commands that may be timing critical and that are used to define the execution of device-dependent operations.(See ANSI X3.130-1986 for a further explanantion and description of these commands).

level 3 (device generic). This term refers to commands that are not timing critical and that are in an intelligent environment in which the slave has functional control (which may or may not be overridden by the master) over the attached facility or facilities.

logical interface. This term refers collectively to all protocols higher than the Physical Interface.

mandatory. To conform to the standard, all functions described as mandatory shall be implemented as defined in this document.

multiplex. This term defines the ability of a master to intersperse the execution of commands between addressees; or of a slave to intersperse the execution of commands between different facilities; or of a slave to intersperse transfer information in bursts that are less than the requested transfer size.

operational. See 4.7.1.3.

optional. This term describes features that are not required by the standard. However, if any feature defined by the standard is implemented, it shall be done in the same way as defined by the standard.

P-available (physically available). See 4.7.1.1.

P-busy (physically busy). See 4.7.1.5.

partition. This term defines a recording area that may be logically addressed. A partition may be slave defined (e.g., data area, CE area, IML area) or may be master defined (e.g., an addressable set of contiguous blocks within the data area). See also *alias*.

PhysicalBlock. This term is uniquely defined in this document as meaning the physical representation of data on the media (e.g., sectors or records on disk and blocks or records on tape). It is used to prevent confusion between industry usage of terms.

physical interface. This term refers to the mechanical, electrical, and bus protocol characteristics specified in ANSI X3.129-1986.

queued. This term refers to the ability of a slave to accept multiple commands per Facility Address from the master and execute them in a sequence according to slave-defined or master-defined algorithms.

ready. This term is used to indicate that a slave or facility can execute its intended functions.

response. This term refers to the response made by a slave to advise the master of the results of a command, or of conditions within the slave.

selection address. This is the address used by the master at the Physical Interface to select a slave, a facility, or both. (This may not be the same as the Actual Address if Synonyms are used.)

synonym. This term describes the ability to redefine the Facility Address of a Facility. There may be more than one synonym to address the same facility.

vendor unique. This term defines those features that can be defined by a vendor in a specific implementation. Caution should be exercised in defining and using such features since they may or may not be standard between vendors.

4. Logical Interface Characteristics

4.1 Operations. The Logical Interface uses a packet structure to transfer Commands from the master to the slave and Responses from the slave to the master. A Bus Exchange at the Physical Interface requires a Bus Control sequence and its associated Ending Status sequence to frame the Information Transfer of Commands, Responses, and data.

The Commands and Responses vary in length and only a single packet of data is transferred for each Information Transfer.

4.1.1 Commands. Commands are issued by the master to instruct the slave, facility, or both to perform an operation. The slave returns a Response when the command has been completed (unless inhibited by the "No Response If Successful" attribute set by the master).

4.1.1.1 Command Types. The command types include:

(1) *Control*. The Control commands provide for control of the slave and facility or facilities.

(2) *Position*. The Position commands cause the positioning of the facility or facilities.

(3) *Transfer*. The Transfer commands may cause multiple blocks of data to be transferred between the master and the facility. Before data is actually transferred, the slave activates the Class 2 Interrupt to inform the master that it is ready to transfer data. The complete data transfer may be broken up into several Information Transfers. A slave with Command Queuing shall generate a Transfer Notification to inform the master of the identity of the command for which the transfer is pending.

(4) *Combination*. The Combination commands provide for operations between two facilities attached to the same slave, or between two facilities attached to different slaves if Slave-to-Slave Information Transfers are supported at the Physical Interface. In an operation between two different slaves, the

master shall designate a dominant slave and a subservient slave. The dominant slave assumes the role of the master for the purpose of initiating Information Transfers to carry out slave-to-slave operations.

(5) *Diagnostic.* The Diagnostic commands provide for maintenance and diagnostic operations between a master and a slave or facility. These commands may be product specific, vendor specific, or both.

4.1.1.2 Command Stacking. A slave that can accept more than one command is capable of command stacking. The number of commands that may be stacked is defined by the slave's attributes. When a master attempts to send more commands to a slave than can be accepted, rejection occurs at the Physical Interface. Commands that are stacked may be Individual or Queued.

4.1.1.2.1 Individual. The slave can accept only one operation for every facility under its control (i.e., a queue of one).

4.1.1.2.2 Queued. The master can have more than one operation for every facility concurrently active under control of the slave, and the slave is responsible to execute them. This permits the slave the freedom to optimize the sequence of command execution to enhance performance (e.g., seek ordering algorithms). The master has the ability to override slave optimization via Attributes.

4.1.1.3 Command Execution Order. The order in which commands are executed is controlled by the command modifier bits that are common to all commands. Commands that are identified as Chained, Sequential, or Ordered cannot be intermixed for a given addressee. The last command of a Chain, Sequence, or Order has no encoding of its identity in order to identify it as being the last one — the slave shall be capable of recognizing it as being the last one, rather than treating it as an Individual or Queued command. The execution scenarios possible are:

(1) *Individual/Queued Commands.* Commands other than those labeled by modifier bits as Chained, Sequential, Ordered, or Priority are executed in a slave-dependent order,a facility-dependent order, or both.

(2) *Chained Commands.* Commands are executed in the order received by the slave (first in, first out (FIFO) order) (i.e., a sequence of commands (not necessarily stacked) to a single addressee). The addressee is implicitly reserved as long as any command of the Chain is being executed. If a command in the Chain is unsuccessful, the Chain shall be terminated and the remaining commands are not executed. The Chained modifier encoding is not set for the last command in a Chain. The slave may multiplex operations for other addressees during execution of the Chain.

(3) *Sequential Commands.* Commands are executed in the order received by the slave (FIFO order). There may be more than one addressee in a Sequence and the slave may multiplex other operations during execution of the Sequence. If a command in the Sequence is unsuccessful, the Sequence shall be terminated and the remaining commands are not executed. The Sequential modifier encoding is not set for the last command in a Sequence. There is no implicit Reserve of the addressee beyond the command being executed.

(4) *Ordered Commands.* Commands are executed in the order received by the slave (FIFO order). There may be more than one addressee in an Order, but the slave shall not multiplex other operations during execution of the Order. If a command in the Order is unsuccessful, the Order shall be terminated and the remaining commands are not executed. The Ordered modifier encoding is not set for the last command in an Order. There is no implicit Reserve of the addressee beyond the command being executed.

(5) *Priority Commands.* Priority commands and Priority Chains/Sequences/Orders are indicated by the command modifier bits. Only the first command in a Chain/Sequence/Order shall be designated as Priority, but the Priority shall apply to the entire Chain/Sequence/Order. A Priority command causes the slave to change the order of execution of stacked commands and also causes changes in interpreting the command packet transmission for each addressee.

When a Priority command packet is received for an addressee, any Chain/Sequence/Order that was in the process of being received shall be ended (i.e., the last packet received is interpreted as the last of the Chain/Sequence/Order regardless of the command modifier settings in that packet. This ending applies regardless of whether the slave is operating in a queued or nonqueued environment.

NOTE: This applies only to commands received over the same port.

Priority commands, whether Individual or part of a Chain/Sequence/Order, are executed before non-Priority commands. Priority commands are

executed in Last In First Out (LIFO) order. With the exception of individual ABORT commands, the receipt of a Priority command does not affect the operation of any Individual command or commands in a Chain/Sequence/Order, except for the possible ending noted previously. When a Priority, Individual ABORT command is received, the slave shall suspend the command executing (if possible), and process the ABORT.

4.1.2 Operation Responses. Each Command has an associated Response packet. Command Completion Response packets contain status that notifies the master whether or not the command was successful, and if not successful, why not. The response is not generated or transferred if the "No Response If Successful" attribute is set by the master.

Response packets are also generated to notify the master of commands that are ready for data transfer, or for asynchronous events that have occurred in the slave or facilities.

4.1.2.1 Interrupts. Physical Interface interrupts are used by the slave to request service from the master.

The master may poll any or all of the interrupt classes by setting the appropriate bits in the Request Interrupts octet of the Physical Interface. It is the responsibility of the master to prioritize the interrupts in the event that more than one slave is interrupting.

Optionally, the master may poll the interrupts from the facilities attached to a slave by the Request Facilities Interrupts octet of the Physical Interface if this feature is supported by the slave.

The Physical Interface ATTENTION IN signal is a logical OR of the interrupts from all but the addressee selected on the Physical interface. The interrupts may be enabled and disabled from generating Attention.

The interrupt classes for slaves and facilities are defined in descending order of priority. Slaves shall present interrupts in order of their priority, except when the response stack is full of lower order interrupt responses (i.e., if a Class 2 could be presented because data is available in a buffer, but the stack is full of Class 1 responses because the master has not requested them, the slave shall not respond to a poll for Class 2 interrupts).

An interrupt is cleared when the condition that caused it to be presented is no longer present.

The three classes of interrupts are as follows:

(1) *Class 3 (Critical Status Pending).* This interrupt shall be used to alert the master to events or conditions existing in a slave or facility that require immediate attention from the master and that are not associated with command completion. An Asynchronous response shall be used to alert the master, and the conditions initiating same are defined by the Major Status, associated Substatus, and Extended Substatus (if any).

(2) *Class 2 (Transfer Status Pending).* This interrupt shall be used by slaves to inform the master that data is ready to be transferred. A Transfer Notification to identify the command involved shall be generated by the slave unless the master is selecting facilities and executing Individual commands; or a Paused transfer implies the Transfer Notification.

(3) *Class 1 (Status Pending).* This interrupt shall be used to identify command completion responses, successful or otherwise, of commands issued by the master, and to inform the master of Asynchronous responses that are not critical. The Major Status and associated Substatus, if any, identify either the conditions under which the command terminated or the cause of the Asynchronous response.

An optional use of this interrupt is to report that the slave, which was previously reported Busy during a selection sequence, is no longer Busy. When the interrupt is used in this manner, the master is responsible for recognizing that there is not necessarily a Command Completion or Asynchronous response packet already in the slave. If there is no response packet pending in the slave, the interrupt shall be cleared by a successful selection sequence.

4.1.2.2 Response Types. The response types do not directly correspond to the three interrupt classes. An Asynchronous response may be Class 3 if critical (e.g., power failure warning), or Class 1 if not critical (e.g., transition from Ready to Not Ready).

4.1.2.2.1 Command Completion Response. This response shall be generated by the slave when a command has been completed, unless the response has been disabled by Attributes being set to "No Response if Successful Completion."

The Class 1 interrupt is activated to inform the master that a Command Completion Response is available.

4.1.2.2.2 Transfer Notification Response (Optional). This response shall be generated by a slave capable of stacking

Individual or Queued commands. It is used to identify the command with which the data transfer to follow is associated.

Transfer Notification Responses are not required under the following conditions:

(1) When commands are Individual, and the command was preceded by a facility selection

(2) For continuation of paused data transfers

The Class 2 Interrupt is activated to inform the master that a Transfer Notification Response is available.

4.1.2.2.3 Asynchronous Response. This response shall be generated by the slave to advise the master of an unanticipated event not associated with command completion.

Either the Class 3 or Class 1 Interrupt is activated to inform the master that this response is available (depending upon whether or not it is considered critical by the slave).

4.1.2.2.4 Imbedded Data Response (Optional). This response shall be generated by the slave to send small amounts of data in a parameter field of the response packet. No more than 254 octets can be transferred in this manner.

The objective of this response is to permit peripherals with very low transfer rates and small transfer needs to gather data in a manner that does not interfere with high-performance peripherals. This response shall not terminate the command, and many of them can be received in answer to a single command.

The Class 2 Interrupt is activated to inform the master that an Imbedded Data Response is available.

4.1.2.3 Response Handling. When there is more than one response packet to be transferred by the slave, it shall transfer response packets to the master in the order of interrupt priority. Within an interrupt class, the responses shall be presented in order of command completion.

If facility selection precedes the request for a response packet, the slave shall only transfer the highest priority response among those for the selected facility.

The master shall properly handle any response packet that it receives, even if it is not necessarily the response type that it expected.

The master should not attempt to initiate the transfer of a response packet before interrupts indicate the availability of a response ready to transfer. If the slave has no response packet to transfer upon receiving a request for one, it shall terminate the attempted Information Transfer (without transferring any information) and not post the Successful bit in Slave Status.

When a slave that supports Control of Bus at the Physical Interface is given control, it may elect to transfer a response packet or transfer data.

4.1.3 Physical Interface Error Recovery Considerations. Errors detected by the Physical Interface protocol require recovery action be taken via a Device-Generic protocol. Recovery procedures depend on the packet type (Command, Data or, Response), and on whether or not the error is detected prior to the Slave Status octet being presented.

4.1.3.1 Recovery from Unsuccessful Slave Status Octet. The Slave Status octet contains the Successful Information Transfer bit, which is set to 0 if any errors are detected by the slave prior to transmission of the octet. This includes not posting Successful if the slave recognized bad parity on the Master Status octet.

NOTE: The slave does not change its Slave Status octet contents based on the contents of the Master Status octet.

Following a command transfer, if the Success bit in either the Master or the Slave Status octet is set to 0, the slave shall ignore the command. The master shall retransmit the command.

Following a data transfer, if the Success bit in either the Master or the Slave Status octet is set to 0, the slave shall terminate the command in progress. A response indicating the failure shall be generated.

Following a response transfer, if the Success bit in either the Master or the Slave Status octet is set to 0, the slave shall retain the response in its buffer and attempt retransmittal under the master's control. The slave shall assert the appropriate interrupt upon deselection.

4.1.3.2 Recovery from Bad Parity on the Slave Status Octet. If the master detects a parity error or any other invalid condition on the Slave Status octet, the slave has no way to determine what has happened (as this is presented in the last state before deselection may occur).

Whenever an invalid condition is detected in the Slave Status octet following a command, data, or Transfer Notification packet transfer, the master shall use an ABORT command or a se-

lective reset (either Logical Interface Reset or Slave Reset) to force the slave to discontinue execution of the command.

If an invalid condition is detected in the Slave Status octet following a response packet other than Transfer Notification (typically Command Completion), the master shall retry execution of the entire command because the slave is not aware of its failure, and has released the buffer containing the response packet.

NOTE: In the case of a slave that queues commands, the master may have to issue no more commands until it can match valid responses with all the issued commands in order to identify the one that was unsuccessful.

If an invalid condition is detected in the Slave Status octet following an Asynchronous response packet, the master has no way of advising the slave that the Slave Status octet was not received correctly. Master-specific recovery procedures may choose whether or not to accept the contents of the Information Transfer. If it is a condition that keeps recurring, the slave shall generate another response.

NOTE: Under this circumstance the error recovery action cannot be certain to obtain the same information as was originally presented.

4.2 Operation Sequences. Operation sequences of the logical interface are controlled by the master, since it establishes the Bus Exchanges that transfer information between the master and slave. Optionally, the master may give control of the bus to a slave; actual control of the Information Transfer at the Physical Interface would then be performed by the slave.

The sequence of actions taken to execute operations between master and slave(s) is done as a series of Information Transfers across the Physical Interface. Slave selection and deselection is done as necessary to communicate with the desired addressee.

4.2.1 Slave Procedures. The following procedures are not necessarily supported by every slave. The slave's Attributes indicate those which are supported.

(1) *Facility Selection and Facility Interrupts.* When the slave supports facility selection and interrupts, the master may make use of them to exert control of the facility. The use of the facility interrupts allows the master to determine the facility requiring service and the facility selection allows the master to direct an operation to the desired facility.

(2) *Slave Control.* When the slave Control of Bus is supported (Bus Acknowledge octet), the master may give control of the subsequent Information Transfer to the slave to allow it to determine whether to transfer a response packet or data. The slave shall not set the Operation Out bits in the Bus Acknowledge octet (i.e., attempt to cause a command to be transferred).

(3) *Queued Facility.* When the slave's Attributes indicate that a facility can operate in a queued environment, the master may issue more than one command for a facility.

4.2.2 Basic Steps. The basic steps performed by the master in carrying out operations are:

(1) Transmit Command Packet To Slave

(2) Poll Interrupts

(3) Receive Response From Slave

(4) Transfer Data Between Master and Slave

These steps shall be as described in 4.2.2.1 through 4.2.2.4.

4.2.2.1 Transmit Command Packet To Slave. The master establishes a Bus Exchange to transfer a single Command packet to the selected slave. The number of commands that can be stacked at an addressee is defined in the Attributes.

Commands consist of variable length transfers, and some slaves may terminate the transfer based on the Packet Length defined in the first two octets. If slaves require master termination of transfers, the length of the transfer is decided by the master.

If the Master Status octet in the Ending Status sequence indicates an unsuccessful transfer, then the slave shall ignore the received command. No response packet shall be generated.

If the Slave Status octet in the Ending Status sequence indicates a successful transfer, the master need not retain the entire command packet.

4.2.2.2 Poll Interrupts. The master polls interrupts to know when to perform data transfers or receive Responses. The polling of a specific interrupt from each of the slaves is performed by the Request Interrupt sequence of the Physical Interface. The presence of an interrupt in any of the slaves is typically provided by the ATTENTION IN signal at the Physical Interface. The master can choose to mask out interrupts by class from setting Attention (via the ATTENTION CONTROL command).

Even if a master has masked out all interrupts from setting Attention, it is still necessary to recognize that unsolicited Attentions could occur (e.g., if an attached slave executed a

Power On reset, the mask information would be lost, and an Attention caused by powering on would be generated when its drivers were enabled.

If facility interrupts are supported by the slave, the master may poll facilities attached to a slave after receiving an interrupt from the slave.

NOTE: The slave generates interrupts on behalf of the attached facility or facilities.

4.2.2.3 Receive Response Packet From Slave. Upon receiving an interrupt from a slave, the master establishes a Bus Exchange to transfer the slave's Response. There is one exception to this action by the master: when the interrupt is Class 2 and the master is executing in a nonqueued environment with facility selection.

The master may give control of the bus to the slave (if this function is supported) so that the slave can decide whether to transfer a response packet or to transfer data.

If the Master Status in the Ending Status sequence indicates a successful transfer, the slave releases the response packet and deactivates the interrupt; otherwise, it retains the packet until successfully transferred.

The response packet may indicate:

(1) The successful or unsuccessful completion of a command

(2) A notice that data is to be transferred for a particular command

(3) The occurrence of an asynchronous event

4.2.2.4 Transfer of Data Between Master and Slave. After receiving a Transfer Notification (or when there is a Class 2 interrupt in a nonqueued environment with facility selection (see 4.2.2.3)), the master initiates a Bus Exchange to transfer data to or from the slave as required by the command identified in the response. (If the slave is given Control of Bus, it sets up the direction of transfer.)

The amount of data intended to be transferred shall be determined by the slave (because it generates SYNC IN at the Physical Interface). Whether or not the ending Master Status indicates a successful input transfer, the slave may release its input data buffer. If the Slave Status indicates a successful output transfer, the master may release its output data buffer.

If the master terminates a data transfer, the characteristics of the slave shall determine whether or not it is an unsuccessful transfer. To a fixed block class of device such as disk, the transfer would be considered unsuccessful unless the transfer was being Paused (optional feature) by the master. To a variable block class of device such as tape, master termination would normally be considered successful and a short PhysicalBlock would be written.

If not Paused by the master, the Command Completion Response shall identify the termination to the master (whether or not the transfer was succesful) as "Unexpected Master Status" in Machine Exception Substatus.

If the transfer requested by the command was completed successfully, unless overriden by Attributes, the slave shall generate a Class 1 interrupt and a Command Completion Response. If the transfer is unsuccessful, the slave shall generate a Class 1 interrupt and report the appropriate status in the Command Completion Response.

4.2.3 Operation Sequence Examples. The four steps described in the preceding subsections may be used in a number of ways, and can best be illustrated by a set of examples executing the same scenario in different implementations. In these examples, the data transfer for Facility 1 requires two transmissions because the slave paused (either unanticipated, or multiplexed) during transfer.

The examples attempt to define only major steps in the flow (e.g., deselection is implicit). To clarify direction of transfer, the terms "transmit" are used for transfers out to the slave, and "receive" is used for transfers in from the slave. The term "recognize" is used for polling because the master has a choice of alternatives when polling (between slaves, facilities and masking).

4.2.3.1 Example of Facility Selection and Individual Commands. In this example, Transfer commands are sent to Facilities 0 and 1 on Slave 0.

(1) Select Slave 0/Facility 0 and transmit Command packet

(2) Select Slave 0/Facility 1 and transmit Command packet

(3) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(4) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1

(5) Select Slave 0/Facility 1 and perform data transfer until paused by Slave 0

(6) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(7) Poll Slave 0 and recognize Class 2 Interrupt from Facility 0

(8) Select Slave 0/Facility 0 and perform data transfer until terminated by Slave 0

(9) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(10) Poll Slave 0 and recognize Class 1 Interrupt from Facility 0

(11) Select Slave 0/Facility 0 and receive Command Completion Response with "Successful" Status

(12) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(13) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1

(14) Select Slave 0/Facility 1 and perform data transfer until terminated by Slave 0

(15) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(16) Poll Slave 0 and recognize Class 1 Interrupt from Facility 1

(17) Select Slave 0/Facility 1 and receive Command Completion Response with "Successful" Status

4.2.3.2 Example of Facility Selection and Queued Commands. In this example, Transfer commands are sent to Facilities 0 and 1 on Slave 0.

(1) Select Slave 0/Facility 0 and transmit Command packet

(2) Select Slave 0/Facility 1 and transmit Command packet

(3) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(4) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1

(5) Select Slave 0/Facility 1 and receive Transfer Notification identifying command

(6) Perform data transfer until paused by Slave 0

(7) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(8) Poll Slave 0 and recognize Class 2 Interrupt from Facility 0

(9) Select Slave 0/Facility 0 and receive Transfer Notification identifying command

(10) Perform data transfer until terminated by Slave 0

(11) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(12) Poll Slave 0 and recognize Class 1 Interrupt from Facility 0

(13) Select Slave 0/Facility 0 and receive Command Completion Response with "Successful" Status

(14) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(15) Poll Slave 0 and recognize Class 2 Interrupt from Facility 1

(16) Select Slave 0/Facility 1 and receive Transfer Notification identifying command

(17) Perform data transfer until terminated by Slave 0

(18) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(19) Poll Slave 0 and recognize Class 1 Interrupt from Facility 1

(20) Select Slave 0/Facility 1 and receive Command Completion Response with "Successful" Status

NOTE: Applications that implement multiple commands per facility in a nonordered sequence, although not included in this example, require the use of Transfer Notification Packets. This example represents the same scenario as the other examples in this section.

4.2.3.3 Example of Slave Selection. In this example, Transfer commands are sent to Facilities 0 and 1 on Slave 0.

(1) Select Slave 0 and transmit Command packet for Facility 0

(2) Remain selected and transmit Command packet for Facility 1

(3) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(4) Select Slave 0 and receive Transfer Notification identifying command for Facility 1

(5) Perform data transfer until paused by Slave 0

(6) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(7) Select Slave 0 and receive Transfer Notification identifying command for Facility 0

(8) Perform data transfer until terminated by Slave 0

(9) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(10) Select Slave 0 and receive Command Completion Response for Facility 0 with "Successful" Status

(11) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(12) Select Slave 0 and receive Transfer Notification identifying command for Facility 1

(13) Perform data transfer until terminated by Slave 0

(14) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(15) Select Slave 0 and receive Command Completion Response for Facility 1 with "Successful" Status

4.2.3.4 Example of Slave Selection and Slave Control of Bus. In this example, Transfer commands are sent to Facilities 0 and 1 on Slaves 0 and 1, respectively. This introduces implicit Transfer Notification (i.e., only one data transfer is paused at a slave, so the Transfer Notification can be implied when the transfer is continued).

(1) Select Slave 0 and transmit Command packet for Facility 0

(2) Select Slave 1 and transmit Command packet for Facility 1

(3) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(4) Select Slave 0 and with Bus Control give Control of Bus to slave

(5) Receive Bus Acknowledge octet indicating Response (Operation In)

(6) Receive Transfer Notification identifying command for Facility 0

(7) Master sets Bus Control (Data Direction)

(8) Perform data transfer until paused by Slave 0

(9) Poll interrupts and recognize Class 2 Interrupt from Slave 1

(10) Select Slave 1 and with Bus Control give Control of Bus to slave

(11) Receive Bus Acknowledge octet indicating Response (Operation In)

(12) Receive Transfer Notification identifying command for Facility 1

(13) Master sets Bus Control (Data Direction)

(14) Perform data transfer until terminated by Slave 1

(15) Poll interrupts and recognize Class 2 Interrupt from Slave 0

(16) Select Slave 0 and with Bus Control give Control to slave

(17) Receive Bus Acknowledge octet with implicit Transfer Notification and Data Direction

(18) Perform data transfer until terminated by Slave 0

(19) Poll interrupts and recognize Class 1 Interrupt from Slave 0

(20) Select Slave 0 and with Bus Control give Control of Bus to slave

(21) Receive Bus Acknowledge octet indicating Response (Operation In)

(22) Receive Command Completion Response for Facility 0 with "Successful" Status

(23) Poll interrupts and recognize Class 1 Interrupt from Slave 1

(24) Select Slave 1 and with Bus Control give Control of Bus to slave

(25) Receive Bus Acknowledge octet indicating Response (Operation In)

(26) Receive Command Completion Response for Facility 1 with "Successful" Status

4.3 Multiplexed Data Transfers (Optional). Different computer system architectures will utilize the IPI, and a wide range of products may be intermixed on a single cable. There are occasions in which a transfer may be terminated for either an unanticipated reason or a deliberate event. The interface provides for both to occur via use of the Pause function at the Physical Interface. Command, and response packet transfers cannot be paused - only data transfers may be paused and continued.

It is necessary that a master be able to control some of the ways in which different products from different vendors coexist. One important area is in ensuring equitable use of the bandwidth. A deliberate event to pause a transfer occurs when a master wishes to prevent a particular peripheral from dominating the use of the interface during Information Transfers.

In an application in which the master requests large data transfers, it is preferred that the master be able to predict the use of the interface. The master can accomplish this by multiplexing transfers between addressees on a predictable basis. The master may define (in Attributes) the maximum number of octets to be transferred in any one data transfer (a burst), even though any single command may itself define considerably more.

A command for a 20K data transfer to a slave with a defined 8K maximum burst size would require three data transfers in order to complete. The slave would pause after each 8K burst and complete after the remainder of 4K was transferred.

4.3.1 Physical Interface Pause and Continue (Optional). The Physical Interface provides the ability to Pause and Continue transfers from both the master and the slave. The master and the slave may use this function to interrupt a continuous data transfer. The slave shall transfer data in bursts no longer than those defined or set in Attributes.

The master cannot indicate any time dependency to the slave, only whether or not the transfer is paused. However, the slave can indicate to the master some idea of the expected time period that the transfer shall be paused. This is done through use of the Time-Dependent Operation bit in the Slave Status octet.

The meaning of the bits shall be as follows:

Pause		Time-Dependent Operation
0	0	No Delay
1	0	Short Delay
1	1	Long Delay
0	1	No More Data

If for some reason both the master and slave present the Pause bit in their respective ending status octets, the slave Pause shall override the master Pause.

In the following descriptions (4.3.2 – 4.3.9), the term P/TDO refers to the setting of the Pause and Time-Dependent Operation bits in the Slave Status octet.

4.3.2 Slave Pause and Master Continue. The master controls the slaves that receive service. The master shall use Attributes to establish the size of transfer, defined as a burst, to be multiplexed. During data transfers, when the burst defined by the master has been transmitted, the slave shall terminate. If information is immediately available to resume transfer with another burst, it shall so indicate by P/TDO=00. This condition can only occur on multiplexed burst boundaries.

If a slave Pauses with no information remaining in its buffer to be transferred, but it anticipates only a short delay before information shall be available, it shall set P/TDO=10. If, however, the delay is expected to exceed the value defined in Attributes, it shall set P/TDO=11.

If it is deselected, the slave shall generate a Class 2 interrupt when it has more information to transfer (immediately when P/TDO=00).

When a slave transferring information pauses at the end of a burst, the master may choose to deselect and poll the other slaves, or remain selected and continue the transfer.

To continue a paused data transfer, the master shall reselect (if necessary), and resumption of the data transfer may or may not be implicit, based on the mode of interface operation.

The slave shall always have a Transfer Notification packet prepared (if they are being used) so that if the master issues a Response request, the slave can respond with same.

4.3.2.1 Implicit Continue. If the master is executing Individual commands with Facility Selection, it can reissue the same Bus Control octet as for the initial transfer.

If the master is executing Queued commands to a slave that can have more than one paused transfer, the master has to receive a Transfer Notification to know which one is being continued. To do so, the master may issue the Bus Control octet for a Response, or permit the slave Control of Bus and the slave shall set up the data transfer of a Response.

4.3.2.2 Explicit Continue. If the master is executing Queued commands to a slave, but only one transfer can be paused per slave, the master may reissue the same Bus Control Octet as for the initial data transfer.

Alternatively, the master may provide the slave Control of Bus, and the slave shall respond with the Data, Direction, and Control of Bus Accepted bits set in the Bus Acknowledge octet. The explicit continue prevents a slave that can have more than one paused transfer and has been given Control of Bus, from starting a second paused transfer.

NOTE: The paused slave is locked to the same transfer command until it is either successfully or unsuccessfully completed. It is possible for a master to issue commands or request responses from a paused slave, but based on internal design a slave may or may not be able to comply. The only way in which a master can free a paused slave is to continue the data transfer or abort the transfer command that is Paused.

4.3.3 Slave Control of Pause and Continue. A slave may choose to pause an data transfer due to an unanticipated delay encountered (e.g., repositioning, error correction). If so, the slave shall terminate the transfer and set P/TDO=11. The master can deselect the slave and await a Class 2 interrrupt to advise that transfer can be continued.

If the master had deselected, when the slave is ready to continue the transfer, it shall generate a Class 2 interrupt. The master shall reselect and set the Control of Bus bit in the Bus Control octet. If the master did not deselect, it can remain in a loop presenting the Bus Control octet, and being rejected by the slave until the slave is ready to continue.

If the master used the Control of Bus bit, the slave shall set the Data, Direction, and Control of Bus Accepted bits in the Bus Acknowledge octet, and transfer shall begin after the master asserts the MASTER OUT signal.

4.3.4 Master Control of Pause and Continue. When a master encounters a temporary interruption, sufficient to stop a data transfer, it shall initiate termination of the transfer and set the Pause bit in the Master Status octet.

The master may choose to deselect or remain selected to the slave.

To continue a Paused data transfer, the master shall reselect (if necessary), along with the same settings in the Bus Control octet used for the original command, then assert the MASTER OUT signal to restart the data transfer.

The slave shall not generate a Class 2 interrupt, but if it encounters any error while Paused, it may generate either a Class 3 or a Class 1 interrrupt.

4.3.5 Uses of Multiplexing. The three basic uses that are made of multiplexing between addressees are described in 4.3.5.1 through 4.3.5.3.

4.3.5.1 One Paused Transfer per Facility. When Facility Selection is used by the master to issue Individual commands, a slave may have up to 16 concurrent data transfers being multiplexed. It is the master's responsibility to manage the multiplexing of information via selection and reselection of the facility.

4.3.5.2 One Paused Transfer per Slave. When Slave Selection is used by the master, a slave may be able to multiplex only one transfer (defined by Attributes). Continuing a paused transfer does not require the use of a Transfer Notification (unless enabled by Attributes) since the master can reissue the same Bus Control octet as for the initial transfer to indicate an implicit Transfer Notification.

4.3.5.3 Multiple Paused Transfers per Slave. When Slave Selection is used by the master, a slave may be able to multiplex more than one transfer. This application requires the use of Transfer Notification packets unless a master remains selected. Upon selection, the slave shall present a Transfer Notification packket to advise the master which paused data transfer is going to be continued.

4.3.6 Anticipated Pause. The master uses Attributes to advise the slave of the burst size to be transferred during a multiplexed transfer.

After the slave has terminated a burst data transfer, it posts Slave Status with the Successful and the requisite P/TDO bits.

(1) *P/TDO=00 - Immediate Continuation Possible*. On Read, next burst is already buffered and ready to transfer. On Write, the buffer is free to accept next burst from master.

(2) *P/TDO=10 - Continuation Possible after Short Delay*. The slave expects a delay of less than the period defined by the master in Attributes.

(3) *P/TDO=11 - Continuation Possible after Long Delay*. The slave expects a delay longer than the period defined by the master in Attributes.

(4) *P/TDO=01 - No More Data*. The entire data transfer is now complete.

The Time-Dependent delay posted by the slave is an indication, and not a guarantee, because many factors are involved (e.g., a Short Delay could actually be a long one if an unforeseen error condition occurred).

4.3.6.1 Master Stays Selected. If the master chooses to stay selected to the addressee, it continues by issuing a Bus Control octet with the same settings as on the original. If the slave cannot continue, it shall reject the Bus Control octet and use encoded Slave Status to advise why.

4.3.6.2 Master Deselects. If the master deselects, the addressee shall assert its Class 2 Interrupt (and Attention if not overridden by ATTENTION CONTROL) when it is capable of continuing the burst transfer.

The actions to be taken are a function of the application (see 4.3.2.1 and 4.3.2.2).

4.3.7 Unanticipated Pauses by the Slave. If the slave encounters a condition that causes interruption of a data transfer (before reaching a burst boundary), it shall terminate the transfer and advise P/TDO=1x in the Slave Status octet (continuation possible after Delay).

NOTE: P/TDO=00 is invalid, because if it could continue immediately, it would not have paused in the first place.

The master may choose to remain selected or not. The same actions that are defined in 4.3.6 for anticipated pauses shall be followed by the slave and the master in order to continue the transfer.

4.3.8 Unanticipated Pauses by the Master. The master cannot pause more than one data transfer to the slave. The next operation to the same addressee has to be the continuation of the paused data transfer.

If a slave cannot accept a master-initiated Pause, it shall set the Not Successful bit in the Slave Status octet.

Close examination of a vendor's specification is recommended because there are many conditions possible that could cause unpredictable results, or require extensive error recovery.

4.3.9 Multiplexed Transfer Mode Identification. The different ways of multiplexing trans-

Table 1
Different Ways of Multiplexing Transfers

Who Paused	Who Continued	Attribute Settings ID/Octet/Bit (Pause)	Attribute Settings ID (Other)
M	M	6B/4/5	
S	M	6B/5/5, 6B/5/4	66, 67
S	S		6B/2/4, 6B/3/4

fers are defined in Attributes. The various combinations shall be as shown in Table 1.

4.4 Data Groupings. The basic unit of recorded data is the eight-bit octet. Octets are combined to form a PhysicalBlock, which is the recording unit on the media, or may be combined to form a DataBlock, which is the master-defined unit of preference. Data transfer between the slave or facility and the master may be in either PhysicalBlocks or DataBlocks.

4.4.1 PhysicalBlocks. The PhysicalBlock size may be preset by the manufacturer of the device, determined by the slave, or by the master by means of the appropriate command (FORMAT for disks and OPERATING MODE or ATTRIBUTES for tapes). Once established, the PhysicalBlock size becomes an attribute of the facility until changed by another command. If a PhysicalBlock size has not been established by the master, the default value preset in manufacture shall be used by the facility.

Each PhysicalBlock on disks and on tapes that support the Update function shall be capable of being written (updated) by a write operation without requiring the master to access or read any adjacent PhysicalBlocks.

For magnetic disks, this is typically provided by separation of recorded fields. One PhysicalBlock may be updated without reading and rewriting any other PhysicalBlock contained in the physical space. If PhysicalBlocks are recorded in a field (e.g., sector) without intervening gaps, then an uncorrectable error in one PhysicalBlock may cause a data integrity exposure to one of the other PhysicalBlocks.

For magnetic tapes, if the Update function is not supported, all data beyond the last write shall be considered invalid.

4.4.2 DataBlocks. The DataBlock size may be predefined by the slave, or it may be set by the master via the appropriate command (FORMAT or ATTRIBUTES for disks and OPERATING MODE or ATTRIBUTES for tapes). If a DataBlock size has not been established by the master, the default value preset by the manufacturer shall be used for the facility.

For disks, a reset or loss of power from the slave or facility shall not cause the DataBlock size to change. Once established, the DataBlock size remains unchanged until a new FORMAT, ATTRIBUTES, or OPERATING MODE command is issued.

For tapes, a reset or loss of power from a slave or facility may cause the DataBlock size to change. The DataBlock size may need to be re-established by the master (see vendor specification).

4.4.3 Extents. An extent is a contiguous number of blocks beginning at a specified starting address. An extent may be as small as one block, or as large as the entire addressable area of a partition (or volume, if absolute addresses are used).

Extents are primarily used in association with transfer and positioning commands. However, they are also used to describe master-definable partitions.

4.4.4 Partitions. These are addressable areas on a volume, and may be either slave defined or master defined.

For disks, each partition is a contiguous range of blocks that begin on a physical boundary (on disks, this is typically a cylinder boundary).

For tapes, a partition is defined as a contiguous addressable area. On tapes that are not preformatted, no extent range is normally possible, and data shall be written from the beginning of a partition. If a tape uses preformatted addressable areas, then a contiguous range of blocks defines a partition.

4.4.4.1 Slave-Defined Partitions

4.4.4.1.1 Data Partition. The slave shall define for each facility the addressable data area known as the default data partition, which is available to the master for data storage. There is only one default data partition per facility.

For tapes, the slave may define additional data partitions that define storage space in addition to the default data partition.

4.4.4.1.2 Maintenance Partitions. The facility may include space that is dedicated for functions other than user data storage. These may include areas for assignment to describe media defects; diagnostic read and write

operations; support functions, such as the storage of internal microcode routines; ans so on. The placement of nonuser data storage areas and the algorithms for accessing them are implementation dependent and shall be described in the vendor specifications.

The slave-defined maintenance areas are available to the master via the OPERATING MODE command.

4.4.4.2 Master-Defined Partitions

4.4.4.2.1 Data Partitions. The master may choose to subdivide the default data partition into smaller ones.

For disks, each partition, other than the default data partition, requires that a Partition ID be associated with any commands that address data (this may be implied by an Alias address).

For tapes, once a partition is addressed, the slave or facility shall continue processing in that partition until a new partition is chosen by use of a Partition Parameter, or until a volume is demounted (if the partition is on the volume). Subsequent commands after a partition change need not supply a partition ID.

4.4.4.2.2 Maintenance Partitions. The master may have the need to define areas that are excluded from normal data accessing. Up to eight maintenance partitions may be defined by the master. These shall be assigned by the slave out of the default data partition.

The master-defined maintenance areas are only accessible via the OPERATING MODE command.

4.4.5 Alternate Data Areas. The slave may map defective PhysicalBlocks to alternate blocks in order to create a defect-free data area for the master. If a PhysicalBlock has been assigned to an alternate data area, it shall be accessible during normal operations in a manner transparent to the master.

4.4.6 Physical Groups. It is possible for a manufacturer to organize a disk with variable block sizes (as in IPI-2 Format 2 in ANSI X3.129-1986) such that there is more than one PhysicalBlock per identification field. The identification field shall be read in order to access any of the PhysicalBlocks within the Physical Group.

4.5 Media Addressing Definitions. IPI disk transfer commands are multiple block transfers across physical boundaries. The addressee positions itself to the Data Address given in the command packet, locates the block, and begins execution of the command. All read and write transfer operations utilize physical (PhysicalBlock) or logical (DataBlock) addresses, unless overridden by the Absolute Addressing modifier in the Data Address parameter. The absolute method of addressing is available for master-specific use when there is a need for a device-unique addressing mode of operation.

IPI tape transfer commands may be multiple block transfers across physical boundaries. Both explicit positioning (using a Data Address) and implicit positioning are supported. If a Data Address is supplied in the transfer command and the addressee supports explicit positioning for transfer commands, the addressee positions itself to the Data Address in the command packet, and begins execution. Tapes that support implicit positioning begin execution at the next recorded element following the last operation. The command modifier determines which recorded element is next since direction may be changed between any two commands.

For tapes which support both types of positioning, the presence of a Data Address field in the Command Extent parameter indicates that the master intends explicit positioning; otherwise, implicit positioning is used.

4.5.1 Absolute Addressing. Absolute addressing uniquely identifies a location by specifying address values that are implementation dependent (as on disks), or by using Position commands to properly position the media (as on tapes). On tape, positioning would typically be accomplished by using the SPACE BLOCK/FILE MARK command to reach the desired position.

For disks, typical values include cylinder address, head address, and sector address. Cylinder addresses run from 0 to C-1, where C is the number of tracks per disk surface; each possible value represents a particular cylinder. Head addresses run from 0 to H-1, where H is the number of read/write heads on the drive; each possible value represents a particular head. Sector addresses run from 0 to S-1, where S is the number of sectors per track; each possible value represents a definite sector, and the sectors are numbered consecutively.

For tapes, values may include track number, relative block number, and the like. When used, track numbers are unique. Block numbers run from 0 to n and may be either PhysicalBlock or DataBlock values.

Absolute addresses are associated with the media, and references may occur across partition boundaries.

4.5.2 Physical Addressing. Physical addressing uniquely identifies a location by specifying address values that are implementation dependent.

On disks, PhysicalBlocks use the same format (typically cylinder, head, sector) as Absolute addressing, but the PhysicalBlock address of a given sector may be different from that sector's absolute address. Factors leading to such differences include partitions, multiple PhysicalBlocks per sector, multiple PhysicalBlocks per identification field, defect mapping, and interleaving.

PhysicalBlock addresses are associated with the partition, and references cannot occur across partition boundaries.

On tapes, PhysicalBlocks use the same format as Absolute addressing (Blocks and File Marks), except that positioning may be explicit with Data Address or implicit (no Data Address). Alternatively, the SPACE BLOCK/FILE MARK command is used to reach the desired position on the media.

4.5.3 Logical Addressing. For disks, logical addressing uses DataBlock addresses to reference data in partitions, within which all DataBlocks are the same size. DataBlocks in each partition are addressable through a linear address space numbered from 0 to n-1, where n is communicated to the master via Attributes.

For tapes, logical addressing uses DataBlock addresses to reference data in partitions. DataBlock sizes may vary within a partiton, and are addressable through a linear address space. The DataBlock address may be augmented by a track reference to speed positioning, which is vendor specific.

Logical addressing permits the master to use DataBlocks that are a multiple or submultiple of PhysicalBlocks. It is the responsibility of the slave to manage the necessary blocking and unblocking of data, and the slave attributes shall identify whether or not it is capable of providing this function. There are many performance criteria that must be considered when DataBlock and PhysicalBlock sizes are different.

LogicalBlock addresses are associated with the partition, and references cannot occur across partition boundaries.

4.5.4 Media Defect Management Considerations. Media defect avoidance is managed in IPI slaves and facilities using several lists. (See 11.3 and 11.4 for command implementation.)

The facility manufacturer's defect list is specified by the device-level interface. It is used to initialize the slave's Permanent defect list the first time a facility is formatted.

For each facility, a slave shall retain information that allows it to avoid defective media in real time. The representation and use of this Working set of defects is slave specific. The combined contents of the Working Permanent and Working Temporary defect lists reflect the condition of the slave's Working set of defects.

Entries in the Working Permanent defect list cannot be removed by any action of the master defined in the IPI command set. Working Temporary defect list entries are removed by a FORMAT command with the Initialize modifier set.

A Suspect Permanent and Suspect Temporary defect list are provided to contain the identity of defects between the time when they are identified to the slave via a WRITE DEFECT LIST command, and the actual substitution of replacement media during a FORMAT or REALLOCATE command. Defect list entries are moved from a Suspect list to the corresponding Working defect list at that time.

Defects identified in a Suspect list are still in the master's DataBlock addressing space. If automatic reallocation is in effect, the Suspect defect lists are always empty.

Entries can also be added to the Working Temporary defect list by specifying their addresses in defect list parameters to FORMAT or REALLOCATE commands.

Any of the defect lists (Working Permanent, Working Temporary, Suspect Permanent, and Suspect Temporary) can be read using the Read Defect List command. Only Suspect Permanent and Suspect Temporary defect lists may be created or appended using the Write Defect List command.

The following example is intended to illustrate the condition of the various defect management structures in a hypothetical slave that implements all of the available features after the following actions:

(1) FORMAT the drive for the first time since its manufacture.

(2) Issue a WRITE DEFECT LIST command with the Permanent modifier set, and a defect list identifying address 5 as defective.

(3) Execute another FORMAT command.

(4) Issue a REALLOCATE command identifying addresses 50 and 150 as defective.

(5) Issue a WRITE DEFECT LIST command with the Temporary modifier set, and a defect list identifying address 10.

Table 2
The Manufacturer's Original Defect List and
the Slave's Master Defect Map

MANUFACTURER'S ORIGINAL FACILITY DEFECT LIST:

Defect Address
120
130

SLAVE'S MASTER DEFECT MAP:

Original Address	Perm/ Temp	Working/ Suspect	New Address
5	Perm	Working	200
10	Temp	Suspect	
50	Temp	Working	201
120	Perm	Working	202
130	Perm	Working	203
150	Temp	Working	204

Table 2 shows an implementation of defect management procedures by the slave. However, this table by no means decribes a complete implementation. The types of addresses indicated are intentionally vague to avoid any assumptions on reallocation technique. In an actual implementation addresses would be expected to be a mixture of DataBlock and Absolute addresses.

4.6 Interface Addressing Definitions

4.6.1 Actual Addresses. At the Physical Interface, a facility has an address that is unique, and is used by the slave to accomplish selection. The IPI Physical Interface Facility Selection restricts addressing to 16 facilities. The IPI Device-Generic Logical Interface permits addressing of up to 255 facilities.

4.6.2 Selection Addresses. The Level 1 interface defines a selection mechanism that allows selection of a slave or facility. Any one of up to 8 slaves on a master, or (optionally) any one of up to 16 facilities on any of the 8 slaves may be physically selected. The address used in this method is called the Selection Address. It contains at least a Slave Address, and may or may not contain the address of a facility (which may or may not be a synonym).

4.6.3 Command Addresses. Device-Generic commands contain a Slave Address and a Facility Address as part of the basic command packet. The Slave Address in the command packet is compared to the value contained in the Selection Address. The Slave Selection Address has a valid range of 0 through 7, and if the Slave Address is not the same in the command, the command is rejected.

If the Level 1 interface utilized slave selection (i.e., there is no facility component in the Selection address), the Facility Address is not compared to the Selection Address. Therefore, the Facility Address can range from 0 through 255 (x'FF'). The value of 255 denotes that the command is addressed to the slave, and not to a facility.

If the Level 1 interface utilized facility selection, then the Facility Address in the Device-Generic command is compared to the address of the facility contained in the Selection Address. In this case, the Facility Address is limited to the range of 0 through 15. If not the same as the Actual Address or a Synonym, the command is rejected.

4.6.4 Facility Address. The Facility Address is the address contained in the command packet. It may be an Actual address, a Selection address, a Synonym address, or an Alias address.

4.6.5 Synonym Addresses (Optional). Without the introduction of Synonyms, a Facility Address of 00-0F would specify a particular physical facility. There are cases in which a master would like to use a Facility Address to reference a different physical facility than would be addressed if the Facility Address in the command were used directly. For instance, if one facility is not operational, it may be advantagous to utilize a different facility in the place of the malfunctioning facility in a way that is transparent to the master's normal operation, thus not impacting operating systems software.

Another situation occurs when the Operating System has predefined the characteristics of a device based on its address. For instance, three disks and a tape may have Actual addresses 0, 1, 2, and 3 assigned at the Physical Interface. However, if the host computer's Operating System assigns disk addresses as 0-7 and tape addresses as 8-F, a Synonym address may be used to reference the tape by Facility Address 8.

Synonyms can also be used to provide pseudo-device-queuing in the Operating System. Many existing Operating Systems do not dispatch more than one I/O request to a device. This is inefficient when the slave to which the devices are attached is capable of optimizing multiple requests to the same device. More than one Synonym address may be assigned to each Actual Address, to overcome the software limitation.

After a Synonym has been set up as equivalent to the Actual address, all Facility Address references thereafter in the command packets may refer to the Synonym as well as the Actual address. The exception to this occurs if an original Actual Address is reassigned as a Synonym.

Synonym addresses are limited to the range of 00-0F if they are to affect the Physical Interface selection. Synonym addresses may be used in the range 00-FE for Logical Interface addressing.

Synonym addressing is optional, and the Attributes of the slave shall indicate whether or not it is supported. Synonym addressing may be supported by the slave at the Physical Interface (00-0F Selection) only, Logical Interface (00-FE) only, or both. If Synonym addressing is not supported, any attempt to invoke this function shall cause a Command Exception condition.

Support of Synonym addresses is an option. If supported, a Synonym address shall be used to map a Facility Address to an Actual address. Whether Synonym addresses are supported or not, Facility Addresses shall be initiallly mapped to Actual addresses in which the same value corresponds to an existing physical facility. Facility Addresses that do not correspond to physical facilities and have not been redefined are invalid.

Synonym addresses are unique from Alias addresses (i.e., the Facility Address can be either a Synonym or an Alias, but not both).

4.6.6 Alias Addresses (Optional). The use of an Alias allows the Facility Address of a command to be mapped to a data partition within a facility (Aliases cannot be used to address maintenance partitions). If the Facility Address of a command is the address that corresponds to an Alias assignment, that command may not have a partition parameter as a part of the command (i.e., a command may not have two partition parameters, whether explicit or implied). (However, Combination Commands may reference multiple facilities, and thus one partition parameter per referenced facility is allowed.)

Partitions permit the master to define more than one addressable data area per physical volume. Reference to these partitions may be made by prefixing every Extent parameter with a Partition parameter, or by assigning an Alias address. The ATTRIBUTES command may be used to assign Alias addresses to each partition. Alias addresses may be assigned in the range of 00-FE.

After Alias addresses have been set up asequivalents to the Partition of a facility (which may be referred to by either an Actual or a Synonym address), all Facility Address references thereafter in the command packets refer to the Partition of that facility. Alias addresses in the range of 00-0F do not affect the selection of the Physical Interface.

Alias addressing is optional, and the Attributes of the slave shall indicate whether or not it is supported. If Alias addressing is not supported, any attempt to invoke this function shall cause a Command Exception condition.

An Alias address maps into a Facility Address and a data partition. The Facility Address that was obtained as a result of an Alias shall then be mapped into a physical facility; thus it could be subjected to an existing Synonym address or map directly to a physical facility. It cannot go through another Alias definition, as that would result in two implicit partition parameters.

Alias addresses are different from Synonym addresses (i.e., the Facility Address can be either a Synonym or an Alias, but not both).

4.6.7 Partition Parameters. A partition parameter can be used with a command that references a facility.

For disks, if no partition parameter is supplied, then the default data partition on the facility is assumed (unless the partition parameter is implicit by an Alias).

For tapes, if no partition parameter is supplied, the current partition shall be used.

Partition parameters on the OPERATING MODE command are required for the master to gain access to maintenance partitions.

Partitions are assigned identification numbers. Partition zero is the default data partition. Fifteen partitions (01-0F) are reserved for identifying maintenance partitions. Partitions 10-FE are used for partitioning of the default data partition. Partition identification FF is reserved.

Partition identification numbers are facility specific. The same partition number on different facilities may not necessarily reference the same portion of each facility.

4.6.8 Communication Addresses (Optional). Communication devices require an extension to the two-level addressing of the IPI. The CONNECT and IDENTIFY commands that provide this extension shall be as defined in the Generic Command Set for Communications document.[1]

4.6.9 Address Examples. To keep track of the relationship between Synonyms, Aliases, and Actual addresses, the slave needs to maintain a table of equivalents. Table 3 illustrates how the relationships can be maintained.

The FF in the second column of Table 3 is used to identify that the facility is not attached (if physical), or not assigned (if logical), and in the third column it identifies that it is not a partition ID reference.

4.7 Slave and Facility Conditions. The manner in which a slave or facility responds to a command is determined by its condition. The condition of a facility is significant only when the slave is in the P-Available condition.

The condition of a slave or facility is affected by its intrinsic as well as its operational characteristics (e.g., a slave that is capable of command queuing may be L-Available when it is active, whereas a slave that can handle only one command at a time would be Not L-Available when active.

Figure 1 illustrates the hierarchy of slave conditions.

4.7.1 Interface Conditions. To assist the reading of the following, the terms "accept" and "execute" are noted in quotations to emphasize their meaning. All conditions are port relevant, except Operational, which is applicable to the slave.

4.7.1.1 P-Available. This condition indicates that the slave is installed and capable of responding to the Physical Interface. This condition implies that the device is properly cabled and is powered on. The P-Available condition is detected at the Physical Interface by responding to the Request Interrupts Sequence with Power On (Bit 3) asserted in the Request Interrupts octet.

4.7.1.2 Not P-Available. This condition indicates that the slave is not installed, is disabled, or is otherwise incapable of responding to the Physical Interface.

4.7.1.3 Operational. This condition qualifies P-Available. The Operational condition indicates that the slave is capable of processing Bus Exchanges. It also indicates that the slave is detected at the Physical Interface, if the slave is able to respond to the selection sequence by asserting SLAVE IN, or if the slave asserts Ready (Bit 5) in response to the Request Slave Interrupts Sequence.

4.7.1.4 Not Operational. This condition qualifies P-Available. The slave is unable to respond to a selection sequence. The Not Operational condition is detected at the Physical Interface if the slave does not set Ready (Bit 5) in response to the Request Slave Interrupts Sequence.

NOTE: The slave is capable of providing a response to a command packet directed to a facility that is Not Operational. A nonoperational condition at the slave will be detected at the Physical Interface (e.g., lack of response to a selection attempt by the master).

4.7.1.5 P-Busy. This condition qualifies Operational and implies that the slave is capable of processing Bus Exchanges but not on this port because it is currently selected or reserved to another port. The P-Busy condition is detected at the Physical Interface if the slave is able to respond to the selection sequence by asserting SLAVE IN, but does not return its bit-significant address, or if the slave asserts Busy (Bit 6) in response to the Request Slave Interrupts Sequence.

4.7.1.6 Not P-Busy. This condition qualifies Operational and indicates that the slave can process Bus Exchanges at this port. Operational is detected if the slave responds to the selection sequence by asserting SLAVE IN and returning its bit-significant address, or if the slave does not assert Busy (Bit 6) in response to the Request Slave Interrupts Sequence.

4.7.1.7 L-Available. This condition indicates that the slave can "accept" a command from the master.

4.7.1.8 Not L-Available. This condition indicates that the slave can "execute" a Bus Exchange from the master, but cannot "accept" a command.

4.7.1.9 L-Busy. This condition indicates that the slave can respond to Bus Exchanges, but is not capable of "executing" a command from the master.

[1] This standard was under development at the time of publication. Contact Secretariat for current status.

Table 3
Table of Equivalents for Slave

Facility Address	Actual or Synonym	Partition ID	Descriptions
00	00	FF	Actual Address
01	01	FF	Actual Address
02	02	FF	Actual Address
03	03	FF	Actual Address
04	FF	FF	Not attached
05	FF	FF	Not attached
06	FF	FF	Not attached
07	FF	FF	Not attached
08	03	FF	Physical Synonym
09	FF	FF	Not attached
0A	FF	FF	Not attached
0B	FF	FF	Not attached
0C	00	FF	Physical Synonym
0D	01	FF	Physical Synonym
0E	01	FF	Physical Synonym
0F	FF	FF	Not attached
10	00	22	Alias
11	02	25	Alias
12	01	42	Alias
13	08	17	Alias (to a Synonym)
14	01	1A	Alias
15	1B	23	Alias (to a Synonym)
16	FF	FF	Not assigned
17	FF	FF	Not assigned
18	FF	FF	Not assigned
19	02	FF	Logical Synonym
1A	01	FF	Logical Synonym
1B	03	FF	Logical Synonym
1C	00	FF	Logical Synonym
1D	FF	FF	Not assigned
1E	FF	FF	Not assigned
1F	FF	FF	Not assigned

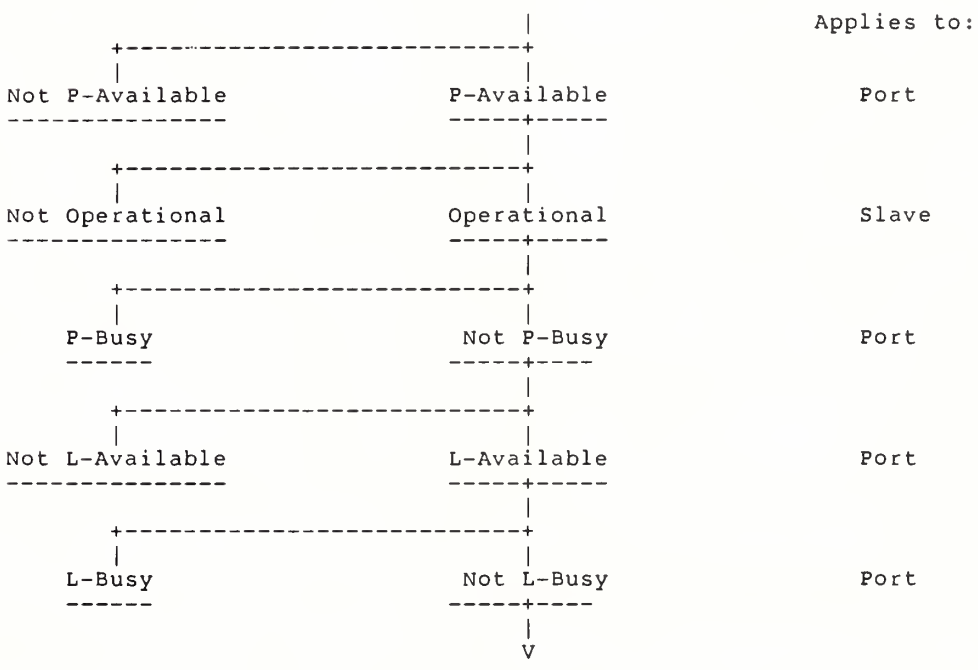


Figure 1
Hierarchy of Slave Conditions

4.7.1.10 Not L-Busy. This condition indicates that the slave can "execute" a command from the master.

4.7.2 General Conditions

4.7.2.1 Active. The addressee has accepted a command, has outstanding status, or both..

4.7.2.2 Inactive. The addressee has no commands or status outstanding.

4.7.2.3 Status Pending. The addressee has status (either asynchronous or in response to a command), to send to the master.

4.7.2.4 Reset. The addressee is in an initial condition in which it has no cognizance of past events. This condition can come about as a result of an external reset by the master, an internal initialization procedure (e.g., following a power-on sequence) or an unsuccessful internal recovery attempt from a severe error.

4.7.3 Operating Status. Slaves provide predictable status on their ability to process commands, especially during a power-on, either at system start or into an operating environment after maintenance.

A slave that is Not Operational is incapable of processing Bus Exchanges, but may be able to communicate what is causing this condition. At the Physical Interface, the Request Interrupts octet can be used to obtain Ready status, but Ready has a double meaning.

Within IPI-2, Ready has the connotation that the device is both operational on the interface and ready for data (e.g., if it is a removable media device, the media is mounted). When an IPI-2 slave reports Ready status to a Request Interrupts, it is able to process Bus Exchanges (Operational) and can process Data Controls.

Within IPI-3, the conditions Operational and Ready are separated because the slave has no concept of Ready, which is a facility concept. When an IPI-3 slave reports Ready status to a Request Interrupts, it is able to process Bus Exchanges (Operational).

There are three Request Interrupts bits that the master can use to build a matrix of information about the attached slaves. Three polls shall be used to build the matrix of Power On, Ready, and Busy. The possible combinations shall be as shown in Table 4.

4.8 Multiple Ports (Optional). Multiple porting includes both manual and programmed static switching (a single slave or a slave and all of its facilities) and dynamic switching (controlled by the master) between two or more IPI ports.

There may be two or more IPI physical ports at the slave and two or more ports at a facility. Any combination of more than two ports between slave and facility, Enable/Disable controls, and the IPI logical constructs for assigning facilities to a physical port are the minimum configuration for multiple port implementations.

It is assumed that two or more slave ports will be connected to different masters and two or more facility ports will be connected to different slaves, but there is nothing that prevents connection of more than one port to the same master or same slave.

When part of all of the resources of a slave or facility become dedicated to performing tasks for a port, there is an allegiance established between the slave or facility and the port. Unless otherwise stated, a slave or facility may have allegiance to only one port at a time. The following description assumes, but does not require, that data transfers and Response Packets always pass through the port that received the corresponding Command Packet.

4.8.1 Slave Switching. There are many different ways of employing a port switch within a slave; however, as viewed by the master, the slave switches all appear to be either physical or logical.

The following capabilities are represented in Figures 2 and 3:

(1) The State Machine represents handling all nonselected states

(2) The Bus Exchange Logic represents the Bus Control/Ending Status sequence, which may or may not include an Information Transfer

(3) The Logical Protocol Execution Logic represents decoding and executing command packets and formulating response packets as well as controlling the facilities if they are logically distinct from the slave.

(4) The Limited Logical Protocol Execution Logic represents decoding and executing command packets and formulating response packets for at least the Priority Reserve command while the slave is implicitly or explicitly reserved to the other port.

(5) The Full Logical Protocol Execution Logic represents decoding and executing command packets and formulating response packets as well as controlling the facilities if they are logically distinct from the slave.

4.8.1.1 Physical Switch. The Physical switch appears at the Physical Interface. As long as the slave is P-Available, it can exe-

Table 4
Possible Combinations of Request Interrupts Bits

Power	Ready	Busy	Slave Condition
0	x	x	Nothing happening at the slave (Not P-Available)
1	0	1	The slave cannot respond to other than Request Interrupts (e.g., it is Busy running diagnostics) (Not Operational)
1	0	0	The slave can respond to Request Transfer Settings and Request Slave/Facility Interrupts (Not Operational)
1	1	0	The slave is able to process Bus Exchanges (Operational)
1	1	1	The slave can process Bus Exchanges, but is currently Busy and unable to do so on this Port (P-Busy)

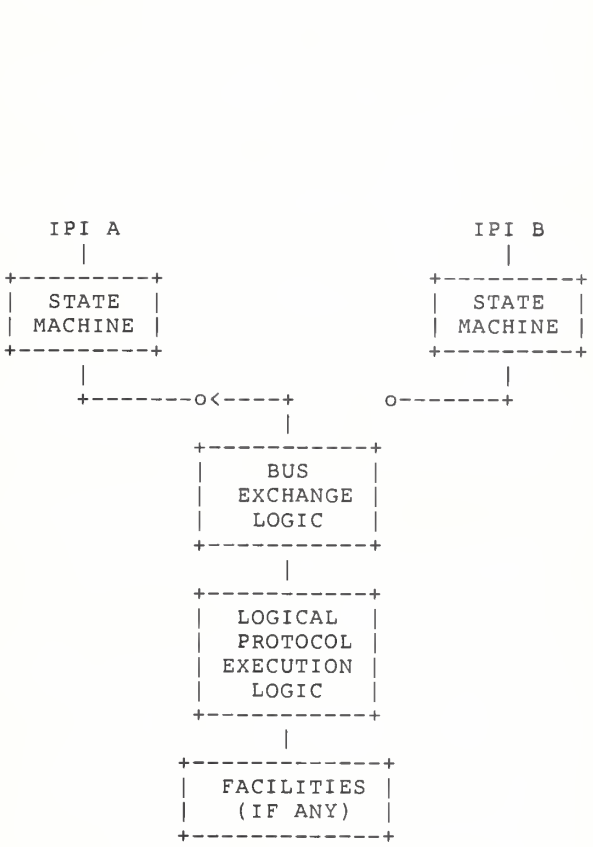


Figure 2
Physical Switch

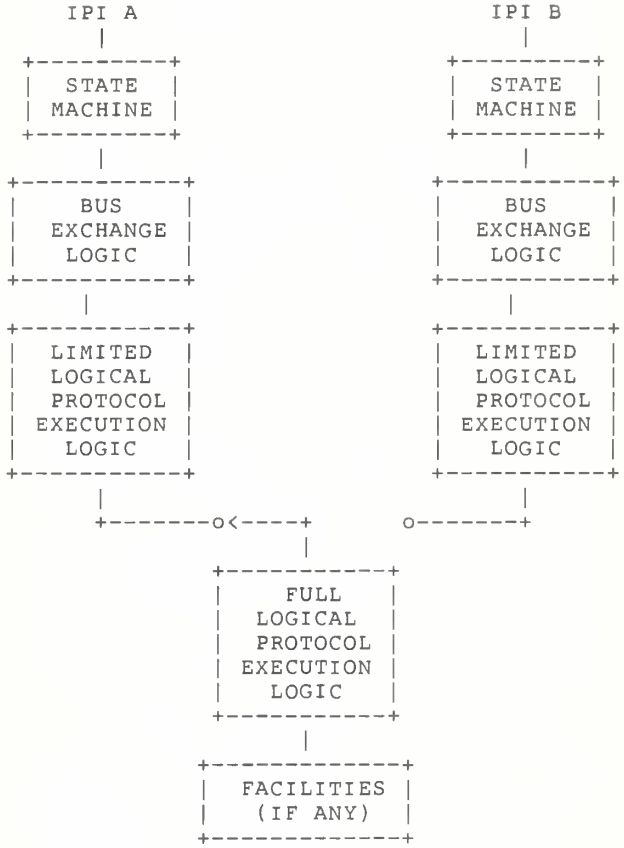


Figure 3
Logical Switch

cute nonselected bus states. When the slave is Not P-Busy, selected bus states can also be executed. When the slave is P-Busy, the master receives a Busy in the Selection sequence.

4.8.1.2 Logical Switch. The Logical switch appears at command decode and execution. When a slave is equipped with this switch, the master may select and transfer commands (provided that the slave has room for the packet), responses (if pending) and data (if the slave is ready for a data transfer).

The Logical Busy (L-Busy) state of the slave may never be sensed by the master since the only effect of the L-Busy state is that command packet execution does not proceed for that port. If the master does become aware of the Busy condition, it is through the command buffer(s) becoming full and command packets being rejected with an appropriate setting in the encoded status field of the Slave Status Octet.

4.8.1.3 Mixed Switch Types. The operation of slaves that employ both Physical and Logical switches is not covered in this standard. Mixing the two types of switches at the slave is implementation dependent, and their interaction is not defined.

4.8.2 Facility Switching. There are many ways to implement a facility port switch, but as viewed by the master, there are only two types of facility switches. The first resides within the slave, where the slave establishes an allegiance between a facility and a particular slave port. The second switch point resides at a logically, and possibly physically, separate facility that can be switched between two or more slaves.

The slave may report the state of the facility switches to the master in several different ways:

(1) Through the slave's own port switch reporting mechanism; that is, Physical switch (P-Busy) or Logical switch (Intervention Required Substatus in a Response packet)

(2) The slave can use the Slave Status Octet in the Bus Exchange

(3) The slave can totally mask the facility switch from the master by managing all of the facility switches itself

In some configurations, there may be both a facility switch at the slave and a facility switch at the facility. For most facility operations, the master cannot distinguish between the two types of facility switching, except for disabling a facility port and performing resets.

4.8.3 Slave Static Switching. Static switching means that a slave and all of its facilities are made P-Available/Not P-Available at an IPI physical port. This is accomplished by manual controls or the PATH CONTROL command. The Enable/Disable controls alter the connection appearance by making it P-Available on only those ports that are enabled. Static switching is implemented at the Physical Interface.

If a manual port control switch is defined for a port, the port control switch has two positions: Enable and Disable. Any port control switch may independently be set to Enable or Disable; thus, any number of ports (including all ports) may be set to Enable or Disable at the same time.

The external form of the port control switch is not specified (e.g., it may be a manually operated switch, or an operator accessible control panel, or a console function). The only requirement is that an operator be able to alter the setting of the port control for a port.

While a slave port is disabled, no signals shall be received by the slave from that port.

Programmed control of the ports is accomplished by the PATH CONTROL command, which provides the capability to Enable and Disable any or all of the ports.

4.8.3.1 Disabling a Slave Port. The manual or orderly command disabling of a slave port takes effect under the following circumstances:

(1) The Physical Interface on that port is in the IDLE state

(2) The slave port control is set to Disable

(3) There are no current or pending commands or responses for any selection address of the port

(4) There are no responses pending for the port

(5) There are no unterminated Chains, Sequences, or Orders for the port, and no facilities assigned to the port are active

A slave port is disabled in an orderly manner either manually or by PATH CONTROL. The slave shall continue to accept requests for the port until all other conditions required for the disabling of the port are satisfied. The acceptance of these requests shall prevent the disabling of the port until the requests are completed. To ensure that the disabling of the port is not delayed indefinitely, the master shall stop sending requests and deselect the slave on the port that has its port control switch set to disable.

A slave port can only be disabled in a de-

structive manner by the PATH CONTROL command. It causes all current or pending commands for the slave port and all nonasynchronous responses to be lost.

4.8.3.2 Enabling a Slave Port. The enabling of a slave port takes effect when its port control is set to Enable and the Physical Interface on that port is in the IDLE state. Any Asynchronous packets to be communicated through a slave port that is disabled shall remain pending until the port becomes enabled or cleared by an appropriate reset. The event of a slave port becoming enabled shall cause an Asynchronous packet to be generated.

Enabling a port shall allow pending Asynchronous packet(s), if any, to be presented to a master. Only the last occurrence of an Asynchronous packet per interrupt class is retained for each addressee unless cleared by an appropriate reset.

4.8.4 Facility Static Switching. Static switching means that a facility is made P-Available/Not P-Available at a slave IPI physical port (either for an individual slave port or for the slave in general) or at the interface between the slave and facility, if present. This is accomplished by manual controls or the PATH CONTROL command.

The Enable/Disable controls alter the connection appearance by making the facility P-Available on only those slave ports or facility ports, or any combination of slave and facility ports, that are enabled. These controls may operate at the slave or at the facility.

If the manual port control switch is defined for any port, the port control switch has two positions: Enable and Disable. Any port control switch may be independently set to Enable or Disable; thus, any number of ports may be set to Enable or Disable at the same time.

The external form of the port control switch is not specified (e.g., it may be a manually operated switch, or an operator-accessible control panel, or a console function. The only requirement is that an operator be able to alter the setting of the port control for a port.

While a facility port is disabled, its Not P-Available condition is signaled in either the Slave Status Octet or the Response Packet, depending on the slave's logical implementation.

Programmed control of the ports is accomplished by the PATH CONTROL command, which provides the capability to Enable and Disable any or all of the ports at both the slave and the logically separate facilities.

4.8.4.1 Disabling a Facility

4.8.4.1.1 Disabling a Facility at a Slave Port. The manual or orderly command disabling of a facility at the slave takes effect when the facility port control for the slave or an individual slave port is set to Disable; the Physical Interface of the slave is not selected on behalf of the facility (Facility Selection); there are no nonstacked, individual, or queued commands for the facility on the port that is being Disabled; there are no responses for the facility pending for the port that is being Disabled; and the facility is not active.

When a facility port control at the slave is set to orderly Disable, the slave shall continue to accept requests from the port until all other conditions required for the disabling of the facility for that port are satisfied. The acceptance of these requests shall prevent the disabling of the port until the requests are completed. To ensure that the disabling of the facility is not delayed indefinitely, the master shall stop sending requests and deselect the slave on behalf of the facility for the port whose port control switch was set to disable.

4.8.4.1.2 Disabling a Facility at a Facility Port. When the facility's port control is separated from the slave, the facility port disable is determined by the slave/facility logical interface. The slave may or may not be able to continue operation until all pending commands and responses have been processed. If the slave cannot continue, the slave may terminate all commands with an appropriate Response Packet until the orderly termination takes effect. If the slave can continue, the disable takes effect at the same time as if the slave had control of the facility switch.

The destructive disabling of a facility port at the slave or facility, performed only with a PATH CONTROL command, causes all command(s) and nonasynchronous responses for the slave port or facility port, or both, to be lost.

4.8.4.2 Enabling a Facility

4.8.4.2.1 Enabling a Facility at a Slave Port. The enabling of a facility port switch at the slave takes effect when the facility's port control is set to Enable and the Physical Interface is not selected on behalf of the facility. Any Asynchronous packets generated at the slave for a facility that is disabled shall remain pending until the port becomes enabled or are cleared by an appropriate reset.

Enabling a facility shall allow pending Asynchronous packet(s), if any, to be presented to a master. Only the last occurrence of an Asynchronous packet per interrupt class shall be retained for each addressee unless cleared by an appropriate reset.

4.8.4.2.2 Enabling a Facility at a Facility Port. The enabling of a facility port switch at a logically separate facility takes effect when the slave/facility interface protocol permits it. The event of a facility port becoming disabled or enabled shall cause the slave to generate an Asynchronous packet for all slave ports at which the facility is enabled.

4.8.5 Slave Dynamic Switching. A slave's dynamic switch may be implemented using either physical or logical switches. In all of these cases, the slave can appear in one of two accessibility modes: Switched or Neutral. Following an internal reset, the slave enters the neutral state.

Interrupts for a port are only presented when the slave is capable of transferring the Response packet associated with the interrupt; however, logical switching does not require the slave to be in the neutral state to transfer response packets.

4.8.5.1 Neutral Mode. In the Neutral Mode the slave has no allegiance to any port. This means that the slave may perform tasks for any port that is enabled.

4.8.5.2 Switched Mode. When a slave enters the Switched Mode, part of the resources of the slave become dedicated to performing operations on behalf of a single port. The slave thus has an allegiance with that port. With the Physical switch, this allegiance is indicated by the other port(s) presenting a P-Busy indication during the selection sequence.

There is not necessarily any indication of what the master can detect with the Logical switch. The master may not be able to determine the cause of the L-Busy condition, or a Bus Exchange that provides Alternate Port Exception Status may be completed.

Whenever an attempt to access a slave is rejected because it is switched to another port (whether P-Busy or L-Busy), the slave shall construct an Asynchronous packet for transmission over the requesting port when the condition that caused the access to be rejected no longer exists. While this Asynchronous packet is pending, if a queued slave accepts a command packet at the port or the slave executes an appropriate internal reset, it shall cancel the pending Asynchronous packet. The generation of the Asynchronous packet, but not the associated interrupt, may be suppressed by Attributes.

4.8.5.3 Implicitly Switched. The slave's port switch becomes implicitly switched whenever the slave starts performing operations on behalf of one port to the exclusion of the other port or ports. For the Physical switch, this means that one port has accepted a selection to the exclusion of the other port or ports. In the Logical switch, this means that command packets are being processed for one port to the exclusion of the other port or ports. The allegiance to the port remains as long as the operations continue on behalf of the port. The allegiance ceases when the operation or series of operations on behalf of the port are completed and the slave is either neutral or performing operations for another port.

Another use of the implicit allegiance is the ability of a slave to maintain an allegiance to a port that has a not Busy interrupt pending. In this case, the slave does not return to neutral when the deselection occurs, but establishes and maintains an explicit allegiance to the interrupting port for the length of time specified in Attributes.

4.8.5.4 Explicitly Switched. Explicit allegiance of a slave with a Physical switch is controlled solely by Priority Select/Priority Hold mechanisms at the Physical Interface. See ANSI X3.129-1986 for specific details.

Explicit allegiance of the slave's Command Execution port switch is established by the PORT ADDRESS command with the Reserve modifier set. The slave establishes an allegiance to the IPI port over which the command packet was received when the command is executed and the port switch is either neutral or switched to the port over which the packet was received. This is not necessarily immediate in an individual stacked or queued environment.

The allegiance ends when the slave has executed the PORT ADDRESS command with the Release modifier set and the Response Packet is transferred to the master or in a Logical switch is stored in a buffer that can be read by the master regardless of the state of the port switch. The allegiance also ends when the slave returns to the neutral mode for reasons other than a PORT ADDRESS command (e.g., reset).

4.8.6 Facility Dynamic Switch. When the dynamic switch is present in a facility or in a

slave on behalf of a facility, it may also appear in either the switched or neutral mode.

Facilities may also be shared between two or more slaves, with different slaves connected to ports of the same facility.

4.8.6.1 Facility Neutral Mode. The slave may communicate on behalf of the facility over any IPI slave port that is enabled.

When an Asynchronous packet (other than one associated with Dynamic Switching) is issued on behalf of a Neutral facility, the packet shall be transmitted over all slave ports that are enabled. The facility condition shall remain as Status Pending until the packet has been transmitted over all ports for which transmission is due.

4.8.6.1.1 Facility Neutral Mode at the Slave Port. Following a slave reset, the facilities affected by the reset become available to all slave ports. The communication over each slave port shall be subject to the same rules as defined for a single port.

4.8.6.1.2 Facility Neutral Mode at the Facility Port. Following a reset of the facility, the facility shall enter the neutral mode and become available to all slaves attached to it. The communication over each facility port shall be subject to the same rules as defined for a single port.

4.8.6.2 Facility Switched Mode. When an attempt to access a facility is rejected because the facility is in the Switched mode, the slave shall construct a Command Completion response. An Asynchronous packet shall be transmitted over the requesting port when the condition that caused the access to be rejected no longer exists, except if it was caused by a reset. While this Asynchronous packet is pending, a queued slave's acceptance of a command packet for the facility shall cancel the pending Asynchronous packet at the port.

While in the switched mode, parts of the facility become dedicated to the slave that caused the switched mode to be entered. The facility may have an implicit or explicit allegiance to any slave whose port is Enabled.

4.8.6.2.1 Facility Switched Mode at a Slave Port. In the switched mode, the facility is associated with a particular IPI port of the slave.

Because of the various types of port switches at the slave, the slave may or may not be able to respond to Bus Exchanges, Information Transfers or combinations of these on behalf of the addressed facility.

4.8.6.2.2 Facility Switched Mode at a Facility Port. In the switched mode, the facility is associated with a particular port of a facility that is logically separated from the slave.

4.8.6.3 Implicitly Switched Facilities. A facility becomes switched to a slave implicitly when the slave initiates execution of a command addressed to the facility.

If the command packet that established the implicit allegiance specified a chain to another command packet, then the implicit allegiance ends when the last command in the chain of commands for that facility has been completed as described previously.

4.8.6.3.1 Implicitly Switched Facilities at a Slave Port. If the command packet that established the implicit allegiance did not specify a chain to another command, the implicit allegiance ends when the Response Packet is prepared to be transferred to the master.

The implicit allegiance may end sooner if the slave, facility, or both are returned to the Neutral mode for some other reason (e.g., reset).

Under these circumstances, a facility may be Status Pending as viewed from one port of a slave and L-Available as viewed from another.

An Asynchronous packet on behalf of a facility pending at a slave port does not establish an allegiance between the slave port and the facility.

4.8.6.3.2 Implicitly Switched Facility at a Facility Port. If the command packet that established the implicit allegiance did not specify a chain to another command, the implicit allegiance at a logically distinct facility ends when the following actions occur:

(1) The operation corresponding to the command packet that established the allegiance is completed by the facility

(2) The information necessary to construct the response packet has been made available to the slave

(3) The slave permits the facility to return to Neutral

The implicit allegiance may end sooner if the facility is returned to the Neutral mode for some other reason (e.g., reset).

Implicit allegiance between the facility and the slave is controlled by the protocols of the slave/facility interface.

4.8.6.4 Explicitly Switched Facilities. Explicit allegiance is established when a PORT

ADDRESS command with the Reserve modifier set is executed by the slave to which the facility is attached. The explicit allegiance ends when a PORT ADDRESS command with the Release modifier set is issued by the master and is executed by the slave to which the facility is attached. The master/slave/facility allegiance continues, however, since the PORT ADDRESS command establishes an implicit allegiance. The allegiance ends as outlined in the subsection on implicit allegiance, 4.8.6.3.

4.8.6.4.1 Explicitly Switched Facilities at a Slave Port. An allegiance is established between the facility and the slave port over which the command packet was transmitted.

4.8.6.4.2 Explicitly Switched Facilities at a Facility Port. An allegiance is established between the slave and the facility port over which the slave communicates with the facility when the command is executed.

4.8.7 Allegiances.

4.8.7.1 Multiple Allegiances. Some applications require the concept of a single master/single slave or single master/single slave/single facility relationship being established with implicit and explicit allegiance for all Information Transfers to be expanded for reasons of performance, resiliency, or both.

4.8.7.2 Explicit Group Allegiance. The master/slave or master/slave/facility allegiance is expanded to include the explicit allegiance of a slave or slave/facility to a single master connected to multiple ports of a slave, a single master connected to multiple ports of a facility through two or more slaves, or either of the two previous cases with two or more masters operating in a coherent manner.

The explicit allegiance is established and relinquished with the PORT ADDRESS command, just as in the single reserve operations, with the exception that a PORT MASK PARAMETER is used to identify the group of ports over which further Command Packets will be accepted. In this case, the Data Information Transfers and the Command Completion Response Packets always are returned over the port that received the corresponding Command Packet.

4.8.8 Alternate Port Notification of Changes. Any time that a FORMAT command is completed (successfully or otherwise), an Asynchronous packet shall be sent to all the alternate ports that are enabled, except the one over which the command was received. The Asynchronous packet shall report Format Completed status to advise the attached master(s) that reinitialization procedures may be necessary for continued use of the facility.

Similarly, any time that an ATTRIBUTES command (with Load or Save modifier) is successfully completed, and the Attributes affect anything other than those specific to a single port, an Asynchronous packet shall be sent to all the alternate ports that are enabled, except the one over which the command was received. The Asynchronous packet shall report Attribute Update Completed status to advise the attached master(s) that reinitialization procedures may be necessary for continued use of the port.

4.9 Reset

4.9.1 External Reset. Any external reset in the form of a Master Reset or Selective Reset, received over the Physical Interface, can be presented by the master at any time regardless of the condition of the slave or facilities. It shall not affect facilities that are switched to another port. The reset shall cause the facilities switched to the master asserting the reset to return to Neutral in addition to establishing the reset condition at the facilities.

The Selective Reset octet provides for resetting the Physical Interface, the Logical Interface, or the slave (as at Power On).

Following the reset condition, the master anticipates that the slave and the affected facilities will become Operational.

4.9.2 Internal Reset. When a slave internal reset occurs, an Asynchronous packet is generated, indicating the condition of the slave. The status of each facility that is affected by the reset should be broadcast over all ports that are Enabled.

Facility resets associated with an internal reset are handled in a slave-dependent manner. It is the slave's responsibility to properly manage the facilities in light of the alternate facility ports, if they exist.

4.10 Bus Octets. The Bus Octets shall be used as defined in ANSI X3.129-1986, with the additions described here.

4.10.1 Facility Selection and Request Facility Interrupts Octets. The Facility Selection and Request Facility Interrupt octets are optional. When the facility interrupts are supported by the slave, they provide a means for the master to determine which facility is interrupting and

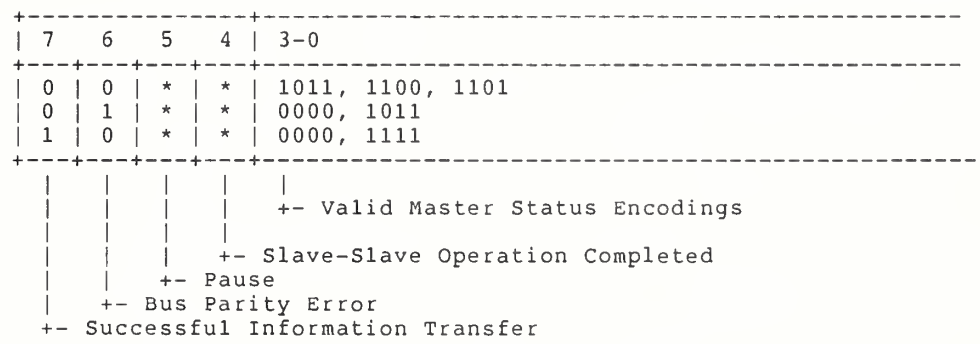


Figure 4
Valid Combinations of the Master Status Octet

the class of interrupt. The facility selection provides a means for the master to direct Information Transfers to a specific facility.

4.10.2 Bus Control Octet. The Bus Control octet of the Physical Interface permits Bits 0-5 to be defined according to the Logical Interface level.

Bit 5 - reserved, set to zero.

Bit 4 - Control of Bus: When Bit 4 = 0, the master shall control the Bus Exchange, with the Bus Control octet defining the subsequent Information Transfer. When Bit 4 = 1, the slave is allowed to control the Bus Exchange. If the slave accepts Control of Bus, the Bus Acknowledge octet shall define the subsequent Information Transfer (the slave's setting shall override that of the master).

Bits 3-0 - set to zero.

4.10.3 Bus Acknowledge Octet. The Bus Acknowledge octet is optional. When it is supported by the slave, it provides a means for the master to turn control of the bus over to the slave, so the slave can determine whether to perform a Response or a data transfer.

Bit 5 - reserved, set to zero.

Bit 4 - Control of Bus Accepted: When Bit 4 = 1, the slave shall define the subsequent Information Transfer by setting the appropriate bits in the Bus Acknowledge octet (which shall override any setting by the master).

Bits 3-0 - set to zero.

Implementation Note: When Bit 4 of the Bus Control octet is set to 0 (master control of Bus Exchange), the slave shall ensure that Bits 0-7 of the Bus Acknowledge octet are set to zero with correct parity.

4.10.4 Master Status Octet. Bits 5-0 of the Master Status octet are defined as optional in the Physical Interface, ANSI X3.129-1986, and their uses are defined in the slave's Attributes.

4.10.4.1 Bit Definitions

Bit 5 - Pause: When Bit 5 = 1, the master is informing the slave that there is more information to transfer in order to complete the just-ended Information Transfer.

Bit 4 - Slave-Slave Operation Completed: When Bit 4 = 1, the dominant slave is noting that the slave-to-slave Information Transfers have been completed (see Physical Interface, ANSI X3.129-1986, for further description of this bit).

Bits 3-0 - Encoded Status: When Bits 0-3 of the Master Status octet are defined by Attributes as being Encoded Ending Status, the following definitions apply. The use of some of these values is applicable only when the following optional capabilities of the Physical Interface are supported:

Control of Bus	1011-1101
Double Octet Mode	1111

0000 No Encoded Status

0001-1010 reserved

1011 Illegal Response Code - the master did not recognize the previously transmitted response packet.

1100 Information Transfer Type Error - the transfer type identified in the Bus Acknowledge octet was incorrect.

1101 Information Transfer Direction Error - the transfer direction identified in the Bus Acknowledge octet was incorrect.

1110 reserved

1111 Odd Octet Transfer - the last transfer of a double octet contained only one octet of information (on BUS A).

4.10.4.2 Valid Combinations. Figure 4 illustrates the valid combinations of the Master Status octet:

```
+---------------+-----------------------------------------------
| 7   6   5   4 | 3-0
+---+---+---+---+-----------------------------------------------
| 0 | 0 | * | * | 0001 through 1100
| 0 | 1 | * | * | 0000, 1111
| 1 | 0 | * | * | 0000, 1111
+---+---+---+---+-----------------------------------------------
  |   |   |   |   |
  |   |   |   |   +- Valid Slave Status Encodings
  |   |   |   |
  |   |   |   +- Time Dependent Operation
  |   |   +- Pause
  |   +- Bus Parity Error
  +- Successful Information Transfer

   * indicates the bit setting may be either 0 or 1.
```

Figure 5
Valid Combinations of Slave Status Octet

4.10.5 Slave Status Octet

4.10.5.1 Bit Definitions

Bit 5 - Pause: When Bit 5 = 1, the slave is informing the master that there is more information to transfer in order to complete the just-ended Information Transfer.

Bit 4 - Time-Dependent Operation: When Bit 4 = 1, the slave is informing the master that the Operation requested in the just-ended Information Transfer will take enough time to complete that deselection may be desirable.

Bits 3-0 - Encoded Slave Status: When Bits 0-3 of the Slave Status octet are defined by Attributes as being Encoded Slave Status, the follow- definitions apply. The use of some of these values is applicable only when optional capabilities of the Physical Interface are supported:

Facility Selection	0100
Data Streaming	1001
Double Octet Mode	1111

In some slave implementations it may not be possible to respond with some of the defined status conditions (e.g., a slave may implement a modular design such that it can accept a command into a buffer and post Successful Information Transfer as a complete and separate function from examining the command, and determining that a Bus Control Reject condition exists. In such an implementation, a Response with associated substatus would have to be presented.

0000 No Encoded Status

0001 Bus Control Reject because it conflicts with current Command context, or because there is no response pending.

0010 Bus Control Reject because addressee can accept no more commands (e.g., if only one command can be issued per addressee, that addressee already has an outstanding Command; if more than one command can be issued, the command buffer has overflowed).

0011 Bus Control Reject because an Asynchronous packet that is pending from the addressee shall be transmitted before any other Bus Control can be accepted.

0100 Selected Facility Busy.

0101 Bus Control Reject due to outstanding Interrupt(s).

0110 Bus Control Reject due to an unsupported Bus Control (e.g., Response stack is full because Class 1 Interrupts have not been accepted).

0111 Bus Control Reject due to an illegal facility address.

NOTE: This is required because only slave selection can be verified at the Physical Interface.

1000 Command Out Bus Control Reject due to unsupported command packet length.

1001 SYNC OUT count not equal to SYNC IN count.

1010 Master termination (in some implementations this is an abnormal condition)

1011 Internal Slave Error

1100 Command Out Bus Control Reject due to Intervention Required

1101 reserved

1110 reserved

1111 Odd Octet Transfer - the last transfer of a double octet contained only one octet of information (on BUS A).

4.10.5.2 Valid Combinations. Figure 5 illustrates the valid combinations of the Slave Status octet.

4.10.6 Request Interrupts Octet. The three interrupt classes, listed in order of descending priority, shall be defined as follows:

Class 3 - Critical Status Pending
Class 2 - Transfer Pending
Class 1 - Status Pending

See 4.1.2.1 for description of interrupts.

4.10.7 Selective Reset Control Octet. Any Selective Reset, with Bits 0,1, or 2 set, or without any Bits 0-3 set shall cause the slave to enable its drivers/receivers (e.g., to enable a slave that may have been disabled as the result of a preceding MAINT state).

The Selective Reset Control octet of the Physical Interface (ANSI X3.129-1986) permits Bits 0-3 to be defined according to the Logical Interface level. These bits have been commonly defined by the present command sets as follows:

Bit 0 = 1 - Physical Interface Reset. The slave shall reset the Physical Interface at the port over which the octet was received, and set the port to neutral.

Interface Reset has no effect on any facilities controlled by the slave, and does not reset any explicit reservations or pending Response packets.

Bit 1 = 1 - Logical Interface Reset. The slave shall reset the Logical Interface at the port over which the octet was received, and set the port to neutral. All slave and facility commands (active and queued), and pending responses from the resetting port shall be reset. All facilities with an allegiance to the reset and all neutral facilities shall be reset. All facilities with an allegiance to the reset port, and all neutral facilities shall be reset with a Logical Interface Reset or equivalent (i.e., alternate port(s) at the facility are not affected. Implicit reservations are reset, but explicit reservations are not affected).

Bit 2 = 1 - Slave Reset. The slave shall reinitialize as at Power On, which means that it shall execute its Initial Microprogram Load procedure (if any) and reset itself to an initial functional condition, with no commands active or responses pending. Note that this typically means that, on the basis of slave implementation, all information in the slave will be lost (see 4.9.2).

All facilities with an allegiance to the reset slave and all neutral facilities shall be reset with a Logical Interface Reset or equivalent (i.e., alternate port(s) at the facility are not affected. Implicit reservations are reset, but explicit reservations are not affected).

NOTE: Facility resets, other than Physical Interface Reset and Logical Interface Reset, are handled by the ABORT command.

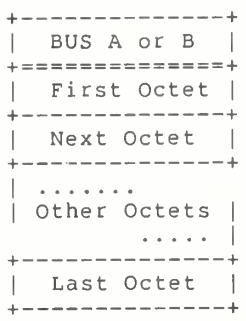


Figure 6
Octet Transfer Positions for Single Octet Mode

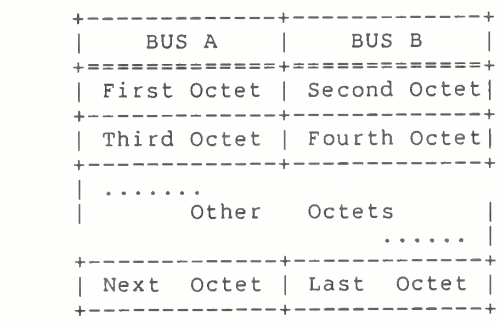


Figure 7
Octet Transfer Positions for Double Octet Mode

Bit 3 = 1 - Slave Release. The slave shall release its interface drivers in the same manner as it would upon recognition of the MAINT state. The slave shall not execute a Reset but the drivers are to remain released on the port over which the reset was received until recognition of another reset.

4.11 ATTENTION IN Signal. The Device-Generic implementation requires that the selected slave not assert ATTENTION IN to indicate the presence of interrupts needing service. It is assumed that the selected slave can use Responses to advise the master of any events that require attention. When a slave is selected, it shall negate its ATTENTION IN signal (if asserted) for the duration of the selection.

4.12 Information Transfers. Information transfers include command packets, reponse packets, and data packets.

4.12.1 Packet Transfer Conventions. Packets are transferred between master and slave in either Single Octet Mode (Figure 6) or Double Octet Mode (Figure 7)(see ANSI X3.129-1986 for further information). In Single Octet Mode, command packets and data are transmitted on BUS A, and response packets and data are transmitted on BUS B. In Double Octet Mode, the first octet of

every pair of the packet is transmitted on BUS A and the other octet is transmitted on BUS B.

4.12.2 Bit Significance Conventions. In the single octet mode (Figure 8) or in the double octet mode (Figure 9), bit 0 is always the least significant bit of the octet and bit 7 is always the most significant bit of the octet.

Octets are transmitted on the bus with their least significant bit (the represented bit 0) as bit 0 of the respective bus (i.e., if an octet is transmitted in Single Octet Mode, the least significant bit of the octet is transmitted as bit 0 of BUS A). If a pair of octets is transmitted in Double Octet Mode, the least significant bit of the first octet (as represented in this document) is transmitted as bit 0 of BUS A and the least significant bit of the other octet is transmitted as bit 0 of BUS B.

4.12.3 Octet Significance Conventions. In the single octet mode (Figure 10) and the double octet mode (Figure 11), the lowest addressed octet of a field is always the most significant octet and the highest addressed octet of a field is the least significant octet.

The most significant octet is transferred first in a multioctet field (e.g., a four-octet field) followed by a two-octet field, as shown in Figure 12.

4.12.4 Command and Response Packet Conventions. Defined fields are fixed in position, even though not all may require valid contents for a particular command or response. Optional-length information, variable-length information, or both shall be presented via parameters that append to the packet-type fields.

Throughout commands, responses, and parameters only single octet fields can begin on an odd octet boundary, all other fields always begin on an even octet boundary, and all values are represented as 16- or 32-bit fields (e.g., a one-octet field and a value needing only 24 bits are not compressed into four octets, but expressed as a one-octet field, a reserved octet, and a four-octet field of 32 bits).

4.12.5 Data Transfer Conventions. Data is transferred between master and slave in either single octet mode or double octet mode. In double octet mode, the octet on BUS A shall always be considered as being first, and the other octet on BUS B shall be second.

Owing to the difference in ordering of the data fields between different computers (micro, mini, mainframe, or vendor dependent), there is no definition of either octet significance of data transferred over the interface.

Figure 8
Bit Positions for Single Octet Mode

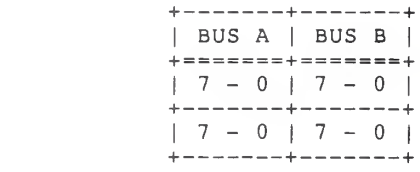


Figure 9
Bit Positions for Double Octet Mode

Figure 10
Positions for Single Octet Mode

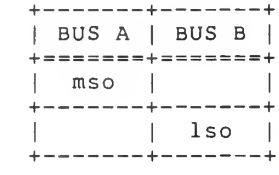


Figure 11
Positions for Double Octet Mode

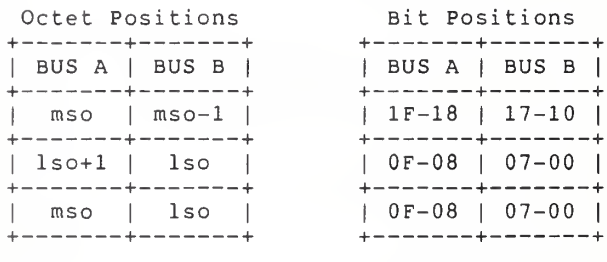


Figure 12
Sequence of Octet Transfers

Figure 13
Parameter Documentation

Data is always transferred in the same sequence, independent of the host computer's data structure. The ordering difference of the host computer can be of major significance if the slave is required to interpret data, as it will have to be cognizant of the host computer's ordering scheme.

5. Message Packet Structure

5.1 Conventions

5.1.1 General Organization. The command and response packets vary in length, and the first octet pair of each packet contains a count of the number of octets in the packet. The next six octets of the command and response packets are identical in format. Both packet types may or may not have a Parameter list that contains non-structured information.

Fields defined within the basic packet (six octets for commands and eight octets for responses) are fixed in position, even though not all may require valid contents for a particular command or response. Conditional or variable information, or both, is presented via parameters appended to the basic packets.

5.1.2 Parameters. Parameters are passed as lists in which each parameter defines its length and is self-identifying. The parameter list is used to pass specific information in the packet that is relevant to the command or response.

Parameters may contain multiple fields and be up to 254 octets in length. Some commands and responses require no parameters appended to the basic packet, and others may have one or more. The length and format of the parameter list varies depending upon the command or response, as well as the ordering, size, and number of parameters. The slave shall verify that all the parameters required by a specific command are present.

Fields in parameters are required to start on an even octet boundary (even boundary octets are transferred on BUS A in DOM), unless they are the second octet of a pair of single octets. Wherever necessary to maintain this rule, a reserved octet shall be used so that the next field shall begin as an even octet. Only single octets and data strings may be represented in other than octet pairs - all numerical values are represented as 16-bit multiples.

A parameter list may contain more than one parameter of the same type. The parameters shall be treated by the slave in the same sequence as that defined by the master. COPY, SEARCH and COMPARE are examples of commands that can make extensive use of the same parameter type multiple times to specify relatively complex command sequences to be executed by the slave.

5.1.2.1 Parameter Documentation. Parameters shall be documented throughout as shown in Figure 13. The @ column is used to identify whether only the master (M), the slave (S), or both (B) can specify the parameter. The Length column contains the legend "$n + 1$" if the parameter is of variable length, or may be reduced in size. The Octet column represents the range of a particular field. The column *X/b* defines whether an octet is encoded or bit significant. Hex values are shown from 00-FF and bit-significant fields are shown as 0-7. The DEF column is used to show defaults.

5.1.2.2 Parameter Length. The Parameter Length shall always begin on a 16-bit boundary, i.e., it shall always be transferred on BUS A of a double octet transfer. It is one octet in size, and the length defined does not include itself.

The value x'00' is a a padding octet in the parameter list that is skipped over and ignored. All masters and slaves that operate with command packets shall be able to handle padding octets, because given the different processors and

micro-processors in use, some implementations may only be able to build a parameter list on boundaries other than octets.

Pad octets, if any, required by the Physical Interface shall be ignored (e.g., in Double Octet Mode, the packets transferred across the Physical Interface are a multiple of two octets, so the master and the slave shall ignore the remaining octet if the basic packet plus the parameters is not an even number.

A parameter may be reduced in size if fields at the end are not required. An effort has been made to place the fields least likely to be needed at the end of parameters. All parameters that may be reduced in size have their Parameter Length noted as "*n* + 1" in the tables.

5.1.2.3 Parameter ID. The identification (ID) of each parameter is specific to the command being executed, unless it is one of those generic to several commands (see 5.5). The Parameter ID is one octet, permitting up to 255 kinds of parameters per command and associated response. Parameter IDs are assigned as follows:

00	Invalid
01	NOP
02	Continuation of Preceding Parameter
03-0F	reserved
10-1F	Slave Major Status
20-2F	Facility Major Status
30-4F	Common Parameters
50-9F	Command Parameters
A0-BF	reserved
C0-CF	Communication Parameters
D0-EF	Vendor-Unique Parameters
F0-FF	Host Adapter Unique Parameters

An ID of x'00' shall be invalid and the command shall be rejected with an Invalid Parameter substatus. An ID of x'01' shall not be processed but ignored and skipped over in the parameter list. An ID of x'02' shall identify this parameter as a continuation of the preceding parameter of 254 octets. This ID cannot be used to continue a parameter that was less than 254 octets. IDs from 10-4F, where assigned, are defined in this section of the document. IDs from 50-9F, where assigned, are command unique and are defined in the command description sections of this document. IDs from C0-CF are reserved for the specific use of Communication commands. IDs from D0-EF are Vendor Unique and, where assigned, shall be defined by the vendor specification. IDs from F0-FF are reserved for use of the host system for any commands that may be interpreted by the Host Adapter, which is responsible for issuing commands over the interface on behalf of the host.

5.1.3 Message Packet Representation in the Document. In Section 5, the packets are shown in a vertical representation that illustrates the physical positioning of fields (see also 4.12). Only the Control command packet is illustrated as it would appear on the interface for both SOM and DOM modes. All other packet representations are only in DOM mode because the information is completely repetitive between the two, and DOM representation occupies less space on the page. The horizontal format used elsewhere in this document is intended to enhance readability, by representing the packets as they would appear in memory.

5.2 Operation Command Packets. Commands to the slave occur as message packets issued to the slave via Information Transfers. There is a limit of one command per Information Transfer. Every command shall cause the slave to return a response when the command has been completed. The response may be implicit at the Logical Interface if the "No Response if Successful" attribute is set by the master. Response packets contain status that notifies the master whether or not the command was successful and, if not successful, why not. Commands to the slave may be Control, Position, Transfer (Primary or Other), Combination Transfer, or Diagnostics. The basic command packet is six octets in length. Parameters may be appended to each command, and the size of the parameter list is included in the Packet Length field. The slave is responsible for checking the fields that apply in a command packet for consistency and validity.

5.2.1 Fields in Command Packets

5.2.1.1 Packet Length. Packet Length contains the actual length (it may be odd or even), of the entire packet, including parameters, expressed in octets. Packet Length does not include the two octets of the Packet Length field itself, nor any null octet required by the Physical Interface in DOM. The slave shall use this value to perform a consistency check on the packet received and to determine the presence of parameters on commands that may or may not require parameters.

Implementation Note: If the Packet Length is used to control a Direct Memory Transfer and not transferred into memory with the packet, some means of recreating it in association with the packet shall be provided.

5.2.1.2 Command Reference Number. The Command Reference Number field contains a value that identifies individual commands from a master. The slave echoes the Command Reference Number in a response packet to identify the associated command. When the slave is capable of queuing multiple commands per addressee, the master is responsible to ensure that all active commands have unique identification. The Command Reference Number is implicitly qualified by the port over which the command was received. Within the slave, the Command Reference Number, plus the Port ID, plus the Slave and Facility Addresses, forms a unique identification number for each command per addressee per port, and the master may only reuse a Command Reference Number when the command is no longer outstanding, i.e., after receiving the response packet for the command.

5.2.1.3 Slave Address. The Slave Address is a value between x'00-07', and is included in the command packet to assist in slave selection error detection. The slave accepting a command packet shall compare its Slave Address against the Slave Address field in the command packet. If the command received contains the wrong Slave Address, the slave shall respond with the appropriate error status. For those commands that may be addressed to either the slave or the facility, the code x'FF' in Facility Address shall identify it as slave only.

5.2.1.4 Facility Address. The Facility Address is a value between x'00-FF'. Unlike the Physical Interface, where Facility Address is restricted by the Select octet to the range 00-0F, the Facility Address in a packet has a valid address range of 00-FE.

The Facility Address FF is used to identify the packet as addressed to the slave only.

There are several ways in which Facility Addresses in the range of 00-FE may be used:

(1) *Actual*. This is the address by which the facility is referenced at the Physical Interface. The address range is limited to 00-0F. The slave shall validate the Facility Address if Facility Selection is being used at the Physical Interface.

(2) *Synonym*. This is an address declared by the master that refers to the facility by other than the Actual address. A Synonym address is specified in Attributes, and may affect both Physical and Logical selection. If a Synonym is declared in the range of 00-0F, the slave may be capable of responding to the Synonym at Physical Interface Facility Selection as well as at Level 3. If a Synonym is declared in the range of 10-FE, it affects only the Logical Interface.

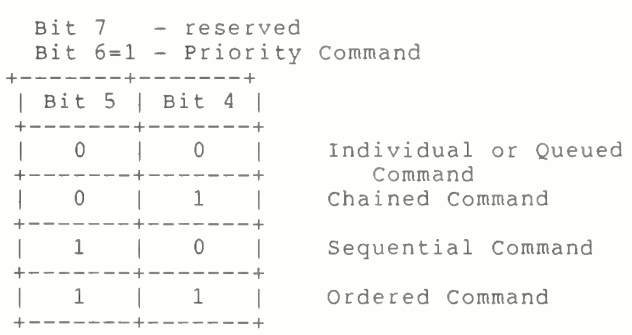


Figure 14
Modifier Octet

(3) *Alias*. This is an address defined by the master to refer to a data partition on a facility (which may be referred to by either an Actual or a Synonym address). An Alias address is specified in Attributes, and may be declared in the range of 00-FE. Note, however, that if a Partition is defined in the range of 00-0F, it has no effect on Physical Interface selection.

5.2.1.5 Opcode. The Opcode field identifies the purpose of the packet and specifies the operation to be performed. A packet's Opcode implicitly determines the interpretation of any parameters that are present.

The Opcodes are interpreted as follows:

00-0F	Control Commands
10-2F	Transfer Commands
30-3F	Combination Commands
40-4F	Position Commands
50-6F	Other Transfer Commands
70-7F	reserved
80-9F	Diagnostic Commands
A0-BF	Communication Commands
C0-DF	reserved
E0-EF	Vendor-Unique Commands
F0-FE	reserved
FF	Asynchronous Response Identifier

5.2.1.6 Modifier Octet. The Modifier Octet is used to express variations in the use or meaning of commands. The Common Modifiers are shown in Figure 14 and defined in 6.1.1.3. They are not required to be supported on every command, but their use is identical on every command in which they are supported (defined by slave attributes).

The Opcode Modifier bits are not defined the same across all commands, and are defined in each command's description (e.g., there is an

SOM Representation:

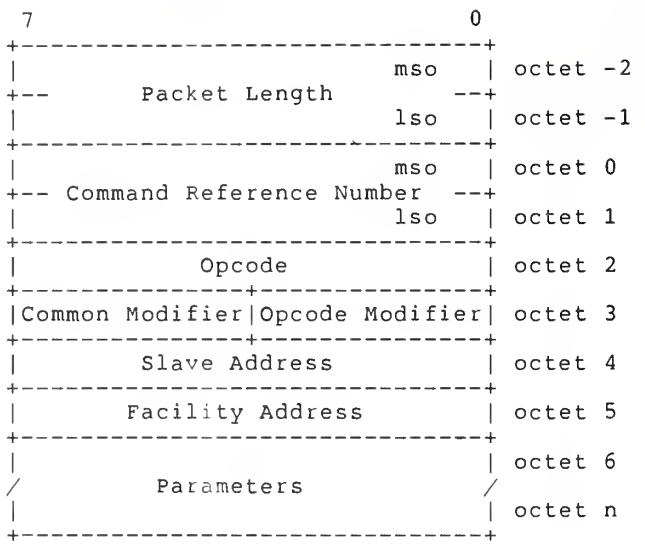


Figure 15
Basic Command Packet

Opcode modifier to distinguish whether the command is to treat the Data Address as applying to DataBlocks or PhysicalBlocks - the same bit position has other meanings on commands that do not require a Data Address.

5.2.1.7 Parameters. The Parameter list is used to pass command-specific information to the slave. Each parameter is self-identifying and the list is unordered unless specified as having to appear in a certain sequence. The presence, length and format of the parameters depend on the command. The length of the parameter list shall be included in the Packet Length.

5.2.2 Basic Command Message Packet. The basic command packet is laid out as shown in Figure 15.

If two parameters were present, the first of five octets (Parameter Length + Parameter ID + a three-octet field), and the second of three octets (Parameter Length + Parameter ID + a one-octet field), it would be a 17-octet transfer, and the parameter list would appear as shown in Figure 16.

If two parameters were present, the first of five octets (Parameter Length + Parameter ID + a three-octet field), and the second of three octets (Parameter Length + Parameter ID + a one-octet field), it would be an 18-octet transfer, and the parameter list would appear as shown in Figure 17.

5.2.3 Command Packet Parameter Requirements

5.2.3.1 Control Command Packet. Control command packets need consist only of the basic packet. There are no required parameters for a control command.

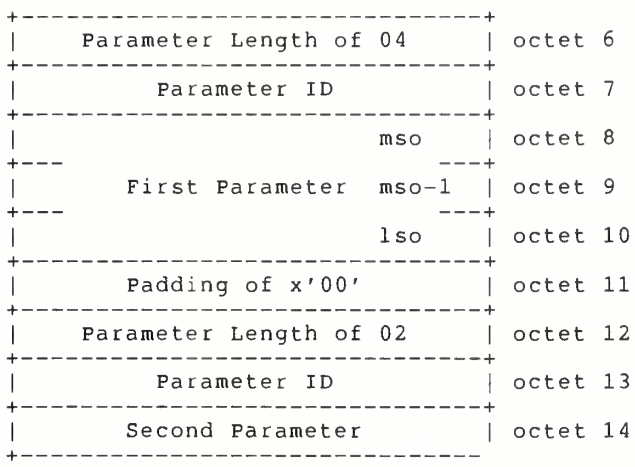
5.2.3.2 Position Command Packet. Unless it is implicit, Position commands require that the Extent parameter be appended to the basic packet.

5.2.3.3 Transfer and Other Transfer Command Packet. Unless it is implicit, Transfer command packets require that the Extent parameter be appended to the basic packet.

5.2.3.4 Combination Transfer Command Packet (Optional). Combination command packets typically require source and destination Combination Extent Parameters. The Extent parameters may be either explicit or implicit and are command specific.

5.2.3.5 Diagnostic Command Packet. Diagnostic command packets typically require parameters.

5.2.4 Transferring Parameters as Data. Parameter lists may be transferred as data with Diagnostic commands such as Read/Write Defect List and Read/Write Error Log. The parameters shall be transferred in as similar a manner as is possible to the way in which the data itself is transferred.



DOM Representation:

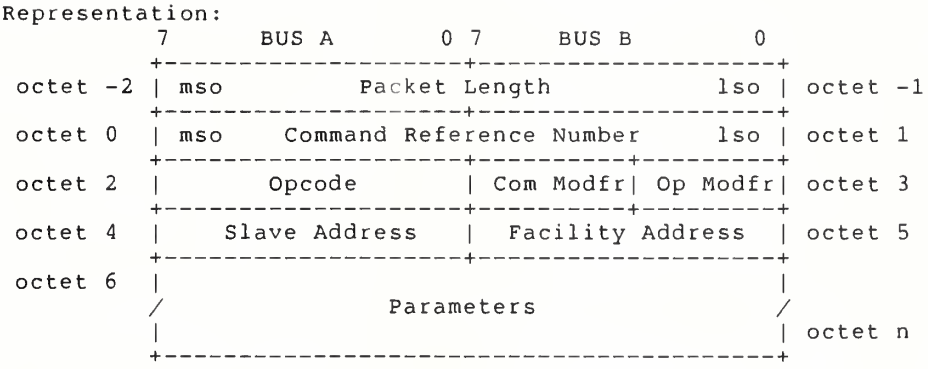


Figure 16
Parameter List for 17-Octet Transfer

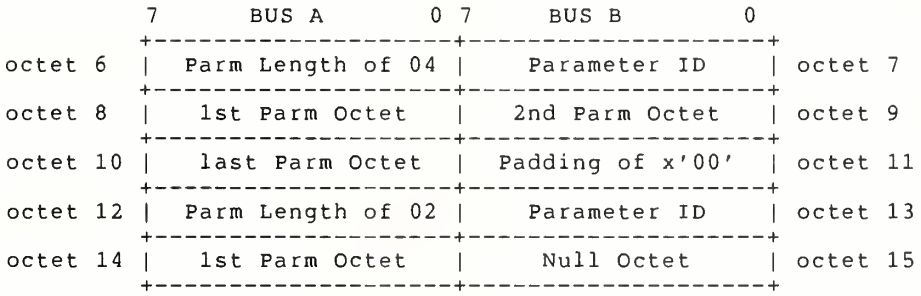


Figure 17
Parameter List for 18-Octet Transfer

The contents of the parameter list may be transferred specifically (by using Parameter ID in appended Request Parameters parameter), or generally (with no specific Parameter ID appended).

When transferred specifically, only one kind of parameter should be transferred as data, since no parameter information is included, and since there is no required ordering of parameters, it may be impossible to know what the contents contain.

When transferred generally, the list may contain different kinds of parameters as they are self-identifying. Parameter lists that exceed 254 octets use continuation parameters. If parameters are transferred by the slave as data in DataBlocks, the slave shall pad with zeros up to the DataBlock size in use.

The master specifies the size of the transfer. If there is a Request Parm Parameter present, it shall immediately precede the Command Extent used for that purpose.

5.2.4.1 Writing. During writing, the master has control of the size of transfer, and uses the Create and Append modifiers.

5.2.4.2 Reading. It is not always possible for the master to predict the size of the parameter list to be transferred when reading. Even when the Request Parm parameter is supported by the slave, the master cannot use it on some commands (e.g., PERFORM DIAGNOSTICS) to find out the length.

The master may issue successive Reads, each containing a Command Extent parameter that identifies the starting address and length to be transferred. The Reads may be Chained or Ordered to prevent intervening commands from being executed while reading parameter data.

Alternatively, the master may request a very high value (as it would to read a tape record of unknown length). The residual of the last Read executed shall define the end of the request information.

Unless requested otherwise by a Request Parms parameter, the slave shall return parameters in a Command Completion response packet if they fit. If they do not fit, the slave shall reject the command as a Missing Parameter, and the master needs to reissue the command with a Request Parms parameter appended to obtain the information as data.

As an example, if data transfers between master and slave are set up to be transferred in DataBlocks of 512 octets, then the list is transferred in as many DataBlocks of 512 octets as is needed. Data transfers continue until as many DataBlocks as are needed to contain the whole list have been transferred. Full DataBlocks shall be transferred, and any remainder per block shall be padded with zeros.

If transfers are not multiplexed, then the slave shall transfer parameter lists up to the limit defined in attributes for data transfers. If the slave contains insufficient buffering or is otherwise unable to transfer the complete list up to the maximum size, then it may use Pause to interrupt the transfer to the master.

If the master needs to limit the size of data transfers, the Count field of the Command Extent parameter may be used to set the number of DataBlocks (or octets depending on the command modifier) to transfer. The maximum transfer size may or may not be less than the length of the parameter list to be transferred. The Address field shall specify the displacement from the beginning of the parameter list.

If more than one transfer is required, then the master should issue multiple commands in a Chain or Order. All data transfers shall be of the size identified in the Command Extent, except the last, which shall terminate the Chain or Order and report a residual in the Response Extent.

NOTE: If the master does not know how much data is to be expected, it has to issue multiple commands and be prepared to have the Chain or Order terminate when the slave has no more parameters to transfer.

5.2.4.3 Example. On a Read Defect List command, the DataBlock size is 512 octets and the slave has 44 entries in the defect list. The residual would be of the number of DataBlocks not transferred; that is, if the command had used the Command Extent parameter with a count of 10, then for both examples the Response Extent would contain 8 as the residual.

5.2.4.3.1 General. The packet length column shown below specifies the length of the parameter list, not the size of the DataBlock.

	Pkt Lth	Parm Lth	Parm ID			
Octet:	0-1	2	3	4-255		
Value:	510	253	56	21*entries		
		256	257	258-509	510	511
		253	02	21*entries	00	00

Octet:	0-1	2	3	4-27	28-511
Value:	26	25	02	2*entries	zeros as padding

Slaves may gather parameters as data in a different manner, depending on implementation. For example, some slaves may begin each DataBlock with a complete parameter and pad out the remainder if a parameter does not completely fit; others may choose to concatenate parameters as tightly as possible, with fields divided over different DataBlocks.

The slave shall ensure that the master receives the same information whether there is single or multiple reads (i.e., if the master requests four DataBlocks in a transfer, the last block shall contain the same information as if one DataBlock had been requested four times).

5.2.4.3.2 Specific. The Request Parm parameter contains the ID 56 and the modifer to transfer Naked Parameters as Data.

Octet: 0-503	504-511
Value: 42*entries	8 octets of 43rd entry

Octet: 0-3	4-15	16-511
Value: 4 octets of 43rd entry	1*entry	zeros as padding

5.3 Operation Response Message Packets. Responses by the slave occur as message packets issued to the master via Information Transfers. There is a limit of one response per Information Transfer.

Responses may advise command completion; be Asynchronous because they are unanticipated events in the slave; or be Transfer Notifications to advise the master of the data transfer about to be initiated.

Response packets contain status to notify the master of the results of a command, or an asynchronous event. The first six octets echo the contents of the basic command packet. The next two octets contain the Response Type and Major Status.

Interrupts are used at the Physical Interface to request the master to read any pending Response packets. Response packets are not necessarily returned to the master in the order of their completion.

5.3.1 Fields In Response Packets

5.3.1.1 Packet Length. See 5.2.1.1

5.3.1.2 Command Reference Number. This field is echoed from the Command packet.

5.3.1.3 Slave Address. This field is echoed from the Command packet.

5.3.1.4 Facility Address. This field is echoed from the Command packet.

5.3.1.5 Opcode. This field is echoed from the Command packet.

5.3.1.6 Modifier. This field is echoed from the Command packet.

5.3.1.7 Response Type. The Response Type is a 4-bit encoded field (bits 4-7 of octet 7) that identifies the packet being presented by the slave. The response shall have the following format:

Bit	Description
0	reserved
1	Standard Command Completion Response Control Position Transfer Other Transfer Diagnostic Not Recognized
2	reserved
3	Extended Command Completion Response Combination Transfer
4	Asynchronous Response
5	Transfer Notification
6	Imbedded Data Response
7-F	reserved

5.3.1.8 Major Status. Major Status is a 12-bit field that tells the master whether the command succeeded or not. All slaves shall return the same status codes for similar situations. The field is bit significant and each Major Status is given a bit code from 0-B (0 corresponds to bit 0 of octet 6, and B corresponds to bit 3 of octet 7). The Major Status categories and their associated Substatus are listed in 5.4.

5.3.1.9 Parameters. The response parameter list is used to pass response-specific data to the master. The presence, length, and format of the parameters are variable. The length of the Parameter field(s) shall be included in the Packet Length. Parameters are self-identifying and are not required to be ordered unless specified as having to appear in a certain sequence.

5.3.2 Basic Response Packet. Basic response packets is laid out as shown in Figure 18.

5.3.3 Response Packet Parameter Requirements

5.3.3.1 Control Response Packet. Control response packets need consist only of the basic packet. There are no required parameters.

5.3.3.2 Position Response Packet. Position response packets need consist only of the basic packet. There are no required parameters.

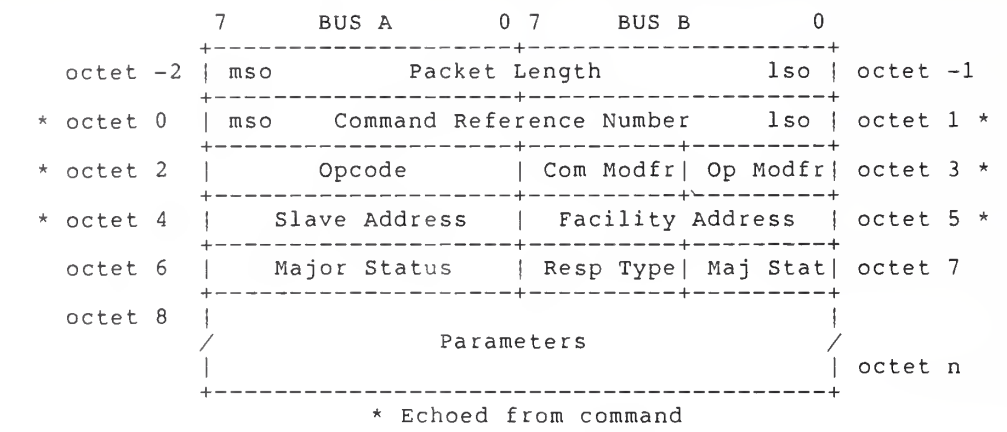


Figure 18
Basic Response Packet

5.3.3.3 Transfer and Other Transfer Response Packet. Transfer response packets need consist only of the basic packet. When a transfer is Not Successful and the extent is explicit, the slave shall append the Response Extent parameter to the basic packet.

5.3.3.4 Combination Transfer Response Packet (Optional). Combination response packets may require two Combination Response Extents, even if successful. The presence, absence, and number of Combination Response Extents required on success or failure is specific to the command.

5.3.3.5 Diagnostic Response Packet. Diagnostic response packets need consist only of the basic packet. There are no required parameters.

5.3.3.6 Asynchronous Response Packet. Asynchronous response packets typically have Substatus parameter(s) appended to the basic packet. This packet is initiated by the slave to advise the master of an unanticipated event.

If the response is not associated with a previously issued command, the Command Reference Number, Opcode, and Modifier are not valid. The Opcode shall contain x'FF', and octets 0-1 and 3 shall be ignored. If the slave can associate the response with a previously issued command, these fields shall be provided to the master. If the asynchronous event is not associated with a facility, octet 5 shall contain x'FF' to identify it as slave-only.

5.3.3.7 Transfer Notification Packet (Optional). This packet is initiated by the slave to advise the master that a data transfer is about to be executed. The Command Reference Number is used by the master to identify which transfer command is to be executed. The Major Status field is unused by this packet, and shall be zero.

It is the slave's responsibility to initiate the data transfer when it is ready to do so by issuing a Transfer Notification Response to prepare the master for the subsequent data transfer. The Transfer Notification Response shall have appended to it any master-specific information that may have been associated with the command by the master as a Transfer Notification parameter (see 5.5.1).

The Transfer Notification packet is required unless the master issues Individual commands with Facility Selection, or if the slave and facility are integrated.

The Transfer Notification packet is always required whenever a READ AT FIRST AVAILABLE DATA command is to be executed, because the Read at First Data parameter is appended. The master shall be provided with this packet prior to accepting any data (see 10.2).

5.3.3.8 Imbedded Data Response Packet (Optional). This packet is used to make small data transfers (typically less than eight octets) in the parameter field attached to the response. Up to 254 octets of data may be appended in this response. More than one Imbedded Data Response packet may be transferred in answer to one command.

Figure 19
Types of Substatus and Responses

5.4 Status

5.4.1 Major Status. Responses may be presented by the slave as the result of a command issued by the master, or may be asynchronous and unexpected as a result of some internal or external condition. Octets 6 and 7 are used to identify the Response Type and the Major Status, and are shown in a horizontal format by bit position in Figure 19. Substatus is specific to a Major Status category and is listed in tabular form following the definitions.

5.4.2 Substatus. There may be more than one Major Status presented (except if Successful). All Major Status categories except Successful have detailed Substatus to identify the specific item that caused its presentation by the slave. This detail is presented in the parameter field that has as its ID in the low-order four bits, the encoded value (0-A) of the Major Status code. Status is reported for the addressee on non-Combination commands or for the dominant slave on Combination commands in accordance with the Substatus listed in Figure 19. If status is presented for subaddresses for Combination commands, it is supplied in the Combination Response Extent (see 5.5.5).

If Extended Substatus is present (as indicated by parameter length), it complements Substatus and is typically used to further define conditions identified in Substatus.

Table 5
Substatus Parameters for Intervention Required

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	14	01-04			INTERVENTION REQUIRED STATUS
		or	01	7		Not P-Available
		24		6		Not Ready
				5		Not P-Available Transition
				4		Not Ready Transition
				3		Physical Link Failure
				2		Attribute Table may be Corrupted
				1		Addressee Busy
				0		
				to		reserved
			04	0		
			05- n			EXTENDED SUBSTATUS (if any)

Table 6
Substatus Parameters for Alternate Port Extension

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	15	01-04			ALTERNATE PORT EXCEPTION STATUS
		or	01	7		Priority Reserve Issued
		25		6		Attributes Updated
				5		Initialization Completed
				4		Format Completed
				3		Facility Switched to Another Port
				2-0		reserved
			02	7		Slave Diagnostic in Process
				6		Slave Diagnostic Terminated
				5		
				to		reserved
			04	0		
			05- n			EXTENDED SUBSTATUS (if any)

5.4.2.1 Intervention Required (ID='x4') (See Table 5). The selected addressee is not able to execute commands, and some intervention is required to make it capable.

5.4.2.1.1 Not P-Available. The selected addressee is not powered on or is not installed.

5.4.2.1.2 Not Ready. The selected addressee cannot execute its intended functions. The addressees' Not Ready condition may be cleared by operator intervention (e.g., a facility made ready by the mounting of a tape volume on a tape unit).

5.4.2.1.3 Not P-Available Transition. This is presented by the slave to advise the master that a facility has become Not P-Available since the time that a command addressed to it was accepted.

NOTE: If the transition had occurred before the command packet was accepted, the status would have been Not P-Available.

5.4.2.1.4 Not Ready Transition. This substatus is presented by the slave to advise the master that a facility has dropped ready since the time that a command addressed to it was accepted.

NOTE: If the transition had occurred before the command packet was accepted, the status would have been Not Ready.

5.4.2.1.5 Physical Link Failure. The slave has detected a catastrophic failure on the external line (e.g., Data Carrier detect Drop on a modem line).

5.4.2.1.6 Attribute Table May be Corrupted. The slave has encountered a condition under which it is possible that the Attributes table has been corrupted, and it is not prepared to continue operation without intervention from the master.

5.4.2.1.7 Addressee Busy. The command cannot be executed because the addressee has been Busy for a time determined by the Facility Timeout Value specified in Attributes.

5.4.2.2 Alternate Port Exception (ID='x5') (See Table 6). The slave or facility has detected an event from an alternate port.

Table 7
Substatus Parameters for Machine Exception

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	16 or 26	01-04			MACHINE EXCEPTION STATUS
			01	7		Addressee No Longer Busy
				6		P-Available Transition
				5		Ready Transition
				4		Operation Timeout
				3		Physical Interface Check
				2		Slave Initiated Reset
				1		Environmental Error
				0		Power Fail Alert
			02	7		Data Check (on Raw Data)
				6		Uncorrectable Data Check (on Perfect Data)
				5		Fatal Error
				4		Hardware Write Protected
				3		Queue Full
				2		Command Failure
				1		
				to		reserved
			03	7		
				6		Read Access Violation
				5		Write Access Violation
				4		Data Overrun
				3		Reallocation Space Exhausted
				2		End of Media detected
				1		End of Extent Detected
				0		Unexpected Master Action
			04	7		Error Log Full
				6		Defect Directory Full
				5		Logical Link Failure
				4		Position Lost
				3-0		reserved
			05- n			EXTENDED SUBSTATUS (if any)

5.4.2.2.1 Priority Reserve Issued. The addressee has been instructed to release allegiance to this port because of a Priority Reserve from an alternate port.

5.4.2.2.2 Attributes Updated. An Attributes command has been issued from an alternate port that has changed the addressee's attributes.

5.4.2.2.3 Initialization Completed. The addressee has completed an initialization procedure that may have affected this port, and was originated by a Reset from an alternate port.

5.4.2.2.4 Format Completed. The addressee has completed a FORMAT command from an alternate port.

5.4.2.2.5 Slave Diagnostic in Progress. The PERFORM SLAVE DIAGNOSTIC command has been issued by an alternate port and is currently in progress.

5.4.2.2.6 Slave Diagnostic Terminated. The slave diagnostic initiated by an alternate port has been completed.

5.4.2.2.7 Facility Switched to Another Port. The slave has determined that the facility is switched to another port.

5.4.2.3 Machine Exception (ID='x6') (See Table 7). Machine Exceptions are the result of a machine condition detected in the slave or an attached facility. Some types of Machine Exceptions are peculiar to an operation (e.g., Data Check exceptions can occur only in connection with operations that transfer data from or to a facility).

5.4.2.3.1 Addressee No Longer Busy. The addressee is notifing the master that it is no longer busy.

5.4.2.3.2 P-Available Transition. This is presented asynchronously by the slave to advise the master that a facility that was previously Not P-Available has become P-Available.

5.4.2.3.4 Ready Transition. This is presented asynchronously by the slave to advise the master that a facility that was not previously ready has become Ready.

5.4.2.3.5 Operation Timeout. There has been a failure condition in the addressee that has been detected by an internal timeout mechanism.

5.4.2.3.6 Physical Interface Check. The slave detected a check condition on the Physical Interface (e.g., invalid sequence generation by the "state machine" or parity error on the bus or buses).

5.4.2.3.7 Slave-Initiated Reset. An internal condition caused the slave to initiate a reset; the master shall assume all outstanding commands and buffer contents are either lost or suspect.

5.4.2.3.8 Environmental Error. Some condition internal or external to the addressee has been detected that may cause a failure condition(s) (e.g., temperature sensor alert).

5.4.2.3.9 Power Fail Alert. The slave has detected an impending power failure condition in itself or an attached facility. The Facility Address field of the Response packet shall contain the address of the affected facility, or x'FF' if it is the slave itself.

5.4.2.3.10 Data Check (on Raw Data). The master has requested raw data and the addressee has detected a data error.

5.4.2.3.11 Uncorrectable Data Check (on Perfect Data). The slave detected a data error that has persisted after the slave has exhausted all possible recovery actions. On write operations, the malfunction may have caused invalid data to be recorded.

5.4.2.3.12 Fatal Error. The addressee detected an internal machine error that precludes execution or continuation of the current command.

5.4.2.3.13 Hardware Write Protected. An attempt was made to write on a facility that was protected against writing by something physical (e.g., Write Protect Ring for tapes).

5.4.2.3.14 Queue Full. The command queue for the addressee is full.

5.4.2.3.15 Command Failure. The command in execution encountered a condition that caused it to complete correctly but unsuccessfully (e.g., a COMPARE of two files detected a discrepancy).

5.4.2.3.16 Read Access Violation. An attempt was made to read on an addressee that had been protected via Access Permits.

5.4.2.3.17 Write Access Violation. An attempt was made to write on an addressee that had been protected via Access Permits.

5.4.2.3.18 Data Overrun. This can occur during direct data transfer, or if the slave has a buffer that was not adequate and the buffer overran during a read or a write operation.

5.4.2.3.19 Reallocation Space Exhausted. Space required for reallocation of data due to media defects is not available; that is, all space assigned for that purpose has been exhausted.

5.4.2.3.20 End of Media Detected. The addressee has detected the end of the media (e.g., the point on tape beyond which the addressee cannot write).

5.4.2.3.21 End of Extent Detected. The addressee has detected the end of the extent (e.g., a File Mark on tape). (On disks, end of partition is detected by a boundary check at command validation and is reported as Command Exception.)

5.4.2.3.22 Unexpected Master Action. The slave has encountered an unexpected action by the master (e.g., Master Status at the Physical Interface does not correlate to the anticipated status, no status was expected and some was presented by the master, the master did not respond with Data In/Out or Control of Bus following a Transfer Notification packet.

5.4.2.3.23 Error Log Full. The capacity of the Error Log has been exceeded.

5.4.2.3.24 Defect Directory Full. The capacity of the Defect Directory has been exceeded and no more blocks can be reallocated.

5.4.2.3.25 Logical Link Failure. The addressee has detected a failure on the communications logical link (e.g., failure of the remote entity to respond to the communications protocol).

5.4.2.3.26 Position Lost. The addressee has lost knowledge of its position on the media (e.g., a tape has not completed a block and has no current reference to use to judge its position).

5.4.2.4 Command Exception (ID='x7') (See Table 8). This status code is used to report invalid or incorrect values in both the basic packet and parameter list (if any). A command is also invalid if the packet was too short, or did not contain all the parameters required by the Opcode. This status is normally presented before command initiation but slaves may be unable to detect some invalid values until after performing some part of the command.

5.4.2.4.1 Invalid Packet Length. The packet length is invalid (e.g., the length of the parameter list plus the basic packet does not equal the packet length).

5.4.2.4.2 Invalid Command Reference Number. The Command Reference Number is invalid (e.g., duplicates a Command Reference Number in a command that is currently active).

5.4.2.4.3 Invalid Slave Address. The Slave Address in the command packet is invalid (e.g., it does not match the selected slave's address).

Table 8
Substatus Parameters for Command Exception

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	17	01-04			COMMAND EXCEPTION STATUS
		or	01	7		Invalid Packet Length
		27		6		Invalid Command Reference Number
				5		Invalid Slave Address
				4		Invalid Facility Address
				3		Invalid Selection Address
				2		reserved
				1		Invalid Opcode
				0		Invalid Modifier
			02	7		reserved
				6		reserved
				5		Invalid Extent
				4		Out of Context
				3		Invalid Parameter(s)
				2		Missing Parameter(s)
				1		Reserved Value Not Equal to Zero
				0		Invalid Combination
			03	7		Not at Initial Position
				6		
				to		reserved
			04	0		
			05- n			EXTENDED SUBSTATUS (if any)

5.4.2.4.4 Invalid Facility Address. The Facility Address in the command packet is invalid.

5.4.2.4.5 Invalid Selection Address. The facility selected at the Physical Interface does not match the facility address supplied in the command packet. The slave shall use the Physical Interface address when reporting Class 1 Interrupts to notify the master of the command packet error. The response packet shall contain the same facility address that was contained in the command packet.

5.4.2.4.6 Invalid Opcode. The command packet contains an invalid or unsupported Opcode.

5.4.2.4.7 Invalid Modifier. The Modifier is invalid or is not supported for the Opcode specified.

5.4.2.4.8 Invalid Extent. The Data Address plus the Count specified in an Extent parameter is not valid for the addressee.

5.4.2.4.9 Out of Context. The slave encounters a command that it considers to be out of context and therefore cannot process it (e.g., a RESUME command without a previous COPY).

5.4.2.4.10 Invalid Parameter(s). One or more of the parameters in the command packet is invalid. An Invalid Parm parameter shall be used by the slave to clarify the error, unless Invalid Extent is posted for a command with a single extent.

5.4.2.4.11 Missing Parameter(s). One or more of the parameters required in the command is not present. A Missing Parm parameter shall be used to identify the one(s) that is missing (e.g., an Extent that is missing from a command that requires it). Each identification of a missing parameter requires a separate Missing Parm parameter.

5.4.2.4.12 Reserved Value Not Equal to Zero. A reserved value defined by the standard does not contain zero.

5.4.2.4.13 Invalid Combination. The addressee has detected that two valid, but mutually exclusive, options have been selected by the master (e.g., 7-Track and Phase-Encoded options on a tape drive).

5.4.2.4.14 Not at Initial Position. The addressee has been instructed to perform an op-eration that is only valid at its initial position (e.g., a tape may only allow a density change at load point).

5.4.2.5 Command Aborted (ID='x8') (See Table 9). If a command is terminated by an ABORT command, this status code is used in the response packet of the aborted command. Any other commands that may be linked to the one ABORTed shall also be terminated. A command need not necessarily be aborted explicitly (e.g., a failure in one command of a Chain, Sequence, or Order shall cause the remaining commands to be aborted).

Table 9
Substatus Parameters for Command Aborted

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	18	01-04			COMMAND ABORTED STATUS
		or	01	7		Command Aborted
		28		6		Command Sequence Terminated
				5		Unexecuted Command from Terminated Sequence
				4		Command Chain Terminated
				3		Unexecuted Command from Terminated Chain
				2		Command Order Terminated
				1		Unexecuted Command from Terminated Order
				0		
				to		reserved
			04	0		
			05- n			EXTENDED SUBSTATUS (if any)

Table 10
Substatus Parameters for Conditional Success

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	19	01-04			CONDITIONAL SUCCESS
		or	01	7		Logging Data Appended
		29		6		Abort Received: No Command Active
				5		Abort Received: Status Pending
				4		Abort Received: Not Operational
				3		Anticipated Error
				2		Anticipated Data Error
				1		Re-allocation Required
				0		Re-allocation Discontinuity (if automatic)
			02	7		Defect Directory Threshold Exceeded
				6		Error Retry Performed
				5		Data Retry Performed
				4		Motion Retry Performed
				3		Data Correction Performed
				2		Soft Error
				1		Release of Unreserved Addressee
				0		Request Diagnostic Control Command
			03	7		Error Log Request
				6		Non-Interchange Volume
				5		Retension Required
				4		End of Media Warning
				3		Statistics Update Requested
				2		Parameter Update Requested
				1		Asynchronous Event Occurrence
				0		Master-Terminated Transfer
			04	7		
				to		reserved
				0		
			05- n			EXTENDED SUBSTATUS (if any)

5.4.2.5.1 Command Aborted. The command to which this response packet is related was ABORTed by the master.

5.4.2.5.2 Command Sequence Terminated. Command was terminated because this command failed to complete successfully.

5.4.2.5.3 Unexecuted Command from Terminated Sequence. The command related to this response packet was not executed but was terminated because a prior command that was sequenced to it failed to complete successfully.

5.4.2.5.4 Command Chain Terminated. Command Chaining was terminated because this command failed to complete successfully.

5.4.2.5.5 Unexecuted Command from Terminated Chain. The command related to this response packet was not executed but was terminated because a prior command that was chained to it failed to complete successfully.

5.4.2.5.6 Command Order Terminated. Command Ordering was terminated because this command failed to complete successfully.

5.4.2.5.7 Unexecuted Command from Terminated Order. The command related to this response packet was not executed but was terminated because a prior command that was ordered to it failed to complete successfully.

5.4.2.6 Conditional Success (ID='x9') (See Table 10). Substatus detail is in the parameter list. This status is used to advise the master that, although the operation completed successfully without Exceptions, certain conditions

were encountered or certain events occurred, or there were changes in slave statistics or parameters (e.g., data retry was needed by the slave, Error Log threshold was exceeded).

5.4.2.6.1 Logging Data Appended. In this response, the slave has appended information that it is advising the master is relevant to be logged. Such information may directly or indirectly be the result of command completion. For example, laser printers that require regular maintenance based on usage may provide the statistics that indicate maintenance action is desirable; a result of the command in execution at the time a particular usage counter overflowed.

5.4.2.6.2 ABORT Received: No Command Active. An ABORT command was issued to an addressee in the L-Available condition, but the referenced command could not be found.

5.4.2.6.3 ABORT Received: Status Pending. An ABORT command was issued to an addressee that has the response status for the referenced command pending (i.e., the command has been completed).

5.4.2.6.4 ABORT Received: Not Operational. An ABORT command was issued to a facility that is Not Operational.

5.4.2.6.5 Anticipated Error. The addressee has detected a condition that may result in future error conditions (e.g., on disk seek, retries were needed).

5.4.2.6.6 Anticipated Data Error. The addressee has detected a condition that may result in future data errors (e.g., successive retries were needed for reading data from the tape or disk.

5.4.2.6.7 Reallocation Required. The addressee has detected a data error condition that requires reallocation action (e.g., an unrecoverable read error).

5.4.2.6.8 Reallocation Discontinuity. The slave has automatically reallocated a block that contained a data error and the reallocated data is now no longer in close proximity to the blocks previously contiguous to it.

5.4.2.6.9 Defect Directory Threshold Exceeded. The threshold within the addressee's Defect Directory has been exceeded, indicating that there is a limited number of entries remaining for adding more defects.

5.4.2.6.10 Error Retry Performed. The addressee has completed the command, but error retry had to be invoked.

NOTE: Error Retry does not include actions associated with data transfer.

5.4.2.6.11 Data Retry Performed. The addressee has completed the command, but data retry had to be invoked (e.g., physical re-read). Data Retry includes all actions associated with the transfer of data.

5.4.2.6.12 Motion Retry Performed. The addressee has completed the command, but motion retry had to be invoked.

5.4.2.6.13 Data Correction Performed. The addressee has completed the command, but data correction had to be applied.

5.4.2.6.14 Soft Error. The slave detected an internal machine error that did not preclude execution or continuation of the current command.

5.4.2.6.15 Release of an Unreserved Addressee. The addressee has received a release command for which there is no reservation.

5.4.2.6.16 Request Diagnostic Control Command. As a result of executing a diagnostic command that provided more information than can be returned by a Response, the addressee is requesting that the master issue a Diagnostic Read Command.

5.4.2.6.17 Error Log Request. The master is requested to capture the contents of the Error Log (which contains manufacturer-dependent information) because the threshold has been exceeded. The log action should be performed in addition to any other master action that is specified.

NOTE: It is not required that all slaves be capable of providing an Error Log, and a master shall not be required to capture same on slaves that provide an Error Log.

5.4.2.6.18 Non-Interchange Volume. The addressee has received a command from the master that would result in creating a volume that may not be readable on another device (e.g., a tape drive may have executed several successive ERASE commands and created a gap that, if increased by another ERASE command, would cause the tape to appear blank when read). This status will continue to be presented until the master initiates a command that does not violate (or compound a violation to) the applicable interchange standard.

5.4.2.6.19 Retension Required. The addressee has successfully completed the operation just completed but has detected that it is losing tension and requires that a POSITION CONTROL command with the Retension modifier set.

5.4.2.6.20 End of Media Warning (EMW). This indicates that the addressee is approaching

the end of the media (e.g., a tape drive has sensed the End of Tape (EOT) marker).

5.4.2.6.21 Statistics Update Requested. There has been a change in meaningful statistics during the execution of this command, and the master is requested to update its Statistics Table (if any).

5.4.2.6.22 Parameter Update Requested. There has been a change in meaningful device parameters during the execution of this command, and the master is requested to update its Statistics Table (if any).

5.4.2.6.23 Asynchronous Event Occurrence. An asychronous event has occurred that may be described further in Extended Status.

5.4.2.6.24 Master-Terminated Transfer. The previous Information Transfer, which had a Master Termination Parameter, is terminated by the master.

5.4.2.7 Incomplete (ID='xA') (See Table 11). This status is used to advise the master that an otherwise successful command did not complete.

If the the Command Reference Number remains active, then the command is one that can be RESUMEd or LINK RESUMEd.

5.4.2.7.1 Command May Be Resumed. The incomplete command remains on the slave's queue, and its Command Reference Number shall remain valid until the command is resumed or aborted.

NOTE: The following status may be reported also if they are encountered by the slave during execution of a complex command on behalf of the master (e.g., when executing a COPY command to multiple reel tapes, the slave encountered EOT).

5.4.2.7.2 COPY Source Space Empty. The addressee could not find the data on the source space.

5.4.2.7.3 Response Packet Truncated. The maximum Information Transfer Length specified by the attributes was exceeded by the response packet, which was truncated at that size, and the response is considered complete by the slave.

5.4.2.7.4 Select Subservient Slave. The Slave-to-Slave transfer in execution requires that the master select a subservient slave to be used by the dominant slave.

5.4.2.7.5 Connect Unsuccessful. The slave issued a CONNECT command to a facility or logical link but was unable to receive a Connect Acknowledge from the remote host.

5.4.2.7.6 Disconnect Unsuccessful. The slave issued a DISCONNECT command to a facility or logical link but was unable to receive a Disconnect Acknowledge from the remote host.

5.4.2.7.7 Connect Identifier Already Assigned. The master has issued a Conection Identify parameter in the CONNECT command that the slave or facility has found to be already assigned to another logical link.

5.4.2.7.8 Link Not Connected. A frame READ or WRITE was issued to a communications link that was not already established via the CONNECT command.

5.4.2.7.9 Beginning of Media (BOM) Detected. The addressee has detected the beginning of the media (e.g., a tape that was reading backwards has run out of tape).

5.4.2.7.10 End of Media Warning (EMW). The addressee is approaching the end of the media (e.g., a tape has sensed the End of Tape (EOT) marker).

5.4.2.7.11 End of Extent Detected. The addressee has detected the end of an extent (e.g., a File Mark on tape).

NOTE: On disk drives, this condition is detected by a boundary check and reported as Command Exception.

5.4.2.7.12 Block Length Difference. The addressee has detected a block with a length that is not equal to the currently defined length (e.g., on tape, a block with incorrect length was read).

5.4.2.7.13 Unrecorded Media. The addressee has detected that no data is recorded on the media (e.g., a tape has read a length of tape that exceeds its maximum allowable gap without finding any data).

5.4.2.7.14 Data Length Difference. The addressee has not transferred all the information specified in a transfer command. On a tape that is writing variable blocks, this shall cause a short block to be written.

5.4.2.7.15 Block Not Found. A specified (either implicitly or explicitly) block address could not be found by the addressee (e.g., a tape drive executing a Position Control command, where a block numbering sequence error was encountered).

5.4.2.8 Successful. The command was successfully completed, and there are no other Major Status posted. There are no Substatus conditions associated with Successful status.

5.4.2.9 Message/Microcode Exception (ID='x3') (See Table 12). This status code is used to indicate that a message is being sent in the response, or that a failure related to microcode has occurred.

Table 11
Substatus Parameters for Incomplete

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	1A	01-04			INCOMPLETE STATUS
		or	01	7		Command May Be Resumed
		2A		6		reserved
				5		COPY Source Space Empty
				4		Response Packet Truncated
				3		Select Subservient Slave
				2-0		reserved
			02	7		Connect Unsuccessful
				6		Disconnect Unsuccessful
				5		Connect Identifier Already Assigned
				4		Link Not Connected
				3-0		reserved
			03	7		Beginning of Media (BOM) Detected
				6		End of Media Warning (EMW)
				5		End of Extent Detected
				4		Block Length Difference
				3		Unrecorded Media
				2		Data Length Difference
				1		Block Not Found
				0		
				to		reserved
			04	0		
			05- n			EXTENDED SUBSTATUS (if any)

Table 12
Substatus Parameters for Message/Microcode Exception

@	LTH	ID	OCTET	X/b	DEF	SUBSTATUS PARAMETERS
S	n+1	13	01-04			MESSAGE/MICROCODE
		or	01	7		Microcode Data Not Accepted
		23		6		Request Master to IML Slave
				5		Slave Unable to IML
				4		Message *
				3		Microcode Execution Error
				2		Failure Message *
				1		Port Disable Pending
				0		Port Response
			02	7		Facility Status
			02	6		
				to		reserved
			04	0		
			05- n			EXTENDED SUBSTATUS (if any)
						* Mutually exclusive parameters

5.4.2.9.1 Microcode Data Not Accepted. The slave did not accept microcode being loaded.

5.4.2.9.2 Request Master to IML Slave. The slave is unable to IML itself and is requesting the master to assist it by downline loading of microcode.

5.4.2.9.3 Slave Unable to IML. The slave is unable to IML itself.

5.4.2.9.4 Message. The slave included a message within Extended Substatus for the master.

5.4.2.9.5 Microcode Execution Error. The slave encountered an error during execution that is within its own microcode.

5.4.2.9.6 Failure Message. The slave encountered a failure condition that resulted in the identification of a Field Replaceable Unit (FRU). The Extended Substatus contains a message in the following format:

05-06 FRU Number
07- *n* ASCII Text

5.4.2.9.7 Port Disable Pending. The addressee has received a manual or programmed Port Disable command that will take effect when the Disable conditions are met.

5.4.2.9.8 Port Response. A port has executed a Port Response command. Appended after the Message/Microcode Substatus parameter is a

Port Mask parameter, which identifies the port that issued the command, and a Response Information parameter (if any), which identifies the information from the port that issued the command.

5.4.2.9.9 Facility Status. The Extended Substatus (5.4.3) includes all of the status provided by the facility (e.g., if an IPI-2 device is attached, then Extended Substatus contains 32 octets of Status and Extended Status of the facility).

NOTE: This is additional information for the master as it is the responsibility of the slave to map facility status, wherever possible, to IPI-3 Substatus.

5.4.3 Extended Substatus. Extended Substatus contains slave-specific or facility-specific data. The contents for each is defined in the vendor's product specification. If a slave is capable of providing Extended Substatus (which is vendor unique), and there is pertinent extended status to present, then it is appended to Substatus in the parameter list (unless overridden by Attributes). The length of the parameter shall implicitly identify if Extended Substatus is present.

5.5 Common Parameters. These parameters are used on many commands and are grouped here as being common in usage across implementations of the Device-Generic command set for different types of devices.

Under each command description, all parameters associated with the command are listed, except for the Transfer Notification parameter, the Invalid Parm parameter, and the Missing Parm parameter. These are not listed because they apply on every command issued, and their repetition serves no useful purpose.

5.5.1 Transfer Notification Parameter (Optional) (See Table 13). A slave may be capable of accepting multiple data transfer commands, and selecting (via some slave-defined or master-defined algorithm), the sequence in which they are to be completed. The Command Reference Number provided in the Transfer Notification Response is the means by which the slave advises which command is about to be executed. If the master provides a Transfer Notification parameter with a command, the slave is required to retain the parameter and echo it back to the master as a parameter appended to the Transfer Notification Response (if any).

The parameter contains the implementation-dependent information of the master, which may be used for a number of different reasons associated with architecture and memory management techniques. One example of the type of information that may be contained in this parameter would be routing information for a system that has virtual memory management.

The contents of this parameter are always echoed back to the master and may or may not be of any meaning to the slave. If a master and slave are designed to complement each other in a specific configuration, this parameter could contain information such as an algorithm which the slave understands to mean that it must properly sequence the transfer to the master. Such uses of this parameter are not defined in this standard, and it is assumed that the contents of the Transfer Notification parameter are transparent to the slave and echoed back to the master.

It is the slave's responsibility to prepare the master for a subsequent data transfer when it is ready to initiate same, by issuing a Transfer Notification Response packet to the master. The Transfer Notification parameter shall be echoed to the master as part of the response packet. It is variable in length because it is master-specific information.

5.5.2 Command Extent Parameter (See Table 14). This parameter identifies the extent to be used for those commands that perform positioning or transfer data. Unless specifically permitted by the command, there is a limit of one extent per command.

The interpretation of these fields is dependent upon the selected device type, the specified command, and the opcode modifier bits. The value is an unsigned binary number of either blocks or octets, depending on the modifier used in the command packet.

5.5.2.1 Count. The Count is interpreted by Position commands as required. On a Transfer Command, the Count identifies the number of octets or blocks requested to be transferred.

5.5.2.2 Data Address. The Data Address specifies the position on the media. On a Transfer Command, this is the point at which to start the data transfer. The Data Address can be expressed in a number of ways, depending on the attributes of the slave. For example:

(1) Cylinder, Head, and Sector (CCHS) for the PhysicalBlock address of disks

(2) Blocks and File Marks for the PhysicalBlock address of tapes

(3) DataBlock Address for the logical addressing of facilities

Table 13
Transfer Notification Parameter

@	LTH	ID	OCTET	X/b	DEF	TRANSFER NOTIFICATION PARAMETER
B	n+1	30	01- n			Master Specific Information

Table 14
Command Extent Parameter

@	LTH	ID	OCTET	X/b	DEF	COMMAND EXTENT PARAMETER
M	n+1	31	01-04			Count
			05-08			Data Address

Table 15
Response Extent Parameter

@	LTH	ID	OCTET	X/b	DEF	RESPONSE EXTENT PARAMETER
S	n+1	32	01-04			Residual Count
			05-08			Data Address

The slave is responsible to perform a boundary check, whenever applicable, on the validity of the extent (Data Address plus Count).

If the Data Address Parameter (see 5.5.11) is used in the parameter list, only the Count field shall be presented.

5.5.3 Response Extent Parameter (See Table 15). This parameter is added to the Response packet for all Position and Transfer commands that required extent information, or did not complete successfully.

5.5.3.1 Residual Count. This field contains the value remaining to be transferred. The value is an unsigned binary number of either blocks or octets, depending on the units used in the corresponding command packet.

Implementation Note: If the transfer command ended in an unrecoverable data error and the Data Recovery modifier was On, the Residual Count shall contain the number remaining to be transferred, including the one in error. The data in error was not transferred.

If the transfer command terminated in an unrecoverable data error and the Data Recovery modifier was Off, the Residual Count shall contain the number remaining to be transferred, but not including the one in error. The data in error was transferred.

If the transfer command terminated with a recoverable data error because the Transfer parameter requested Stop on Data Error, and Data Recovery was On, the Residual Count shall contain the number remaining to be transferred but not including the one on which the data was recovered. The data recovered was transferred.

5.5.3.2 Data Address. This field contains the address of the next block or octet to be transferred consistent with the setting of the Data Recovery modifier bit. There are considerations associated with End of Media, End of Extent, or both, which affect the ability of the slave to place a valid value in this field (e.g., if status is given for End of Media or End of Extent, the address is not valid).

If the Data Address Parameter (see 5.5.11) is used in the parameter list, only the Count field shall be presented.

5.5.4 Combination Command Extent Parameter (Optional) (See Table 16). All combination commands provide the ability to have one or more extents for the source and destination subaddressees. The command packet addresses either the slave or a facility. If the addressee is the slave, it shall execute the operation on behalf of the master. If the addressee is a facility, all operations are restricted to that facility. Each Combination Extent parameter identifies a subaddressee and its extents. In the case where a subaddressee has no address range information (e.g., tape), the Combination Extent parameter contains only octets 00-03. Where there are ad-

dress ranges to be specified, ordered pairs of fields define the Count and Data Address.

There may be more than one address range within a Combination Extent parameter, and more than one Combination Extent parameter (with different Slave and Facility addresses) may be defined for a source or destination. It is possible for the master to create relatively complex combination commands. As an example, to back up disk files, a master may COPY a source file that is spread over more than one extent on more than one facility to a reel of tape on the destination facility.

Not all slaves offer the full functionality of the combination commands, and their abilities shall be defined in attributes.

5.5.4.1 Slave Address. This field is used when the slave in control of the facility is addressed as source or destination.

5.5.4.2 Facility Address. This field is used when the facility is being addressed as source or destination. If the facility is integrated with the slave, this field shall be set to x'FF'. The Facility Address shall match the addressee if the addressee is a facility.

5.5.4.3 Modifiers

(1) *Bit 7.* This bit identifies the parameter as a primary (typically, the source), or a secondary (typically, the destination) extent.

(2) *Bit 3.* This bit establishes the direction of data transfer.

(3) *Bit 2.* This bit identifies whether physical or logical addressing is used.

(4) *Bit 1.* This bit defines whether or not data recovery is required.

(5) *Bit 0.* This bit identifies whether block or octet counts are used.

5.5.4.4 Count. If extent information is required, this field is always the first of an ordered pair, and the pair may be repeated as many times as there are extents.

5.5.4.5 Data Address. This field is always the second of an ordered pair with the Count. If there are multiple extents to be transferred, the ordered pair is repeated as many times as needed.

NOTE: If the Data Address Parameter (see 5.5.11) is used in the parameter list, the ordered pairing that permits multiple extents cannot be used, and the parameter is truncated at Octet 04, followed by ordered pairs consisting of a Command Extent parameter that includes only the Count field followed by the Data Address parameter.

5.5.5 Combination Response Extent Parameter (Optional) (See Table 17). The basic response packet contains the Major Status associated with the slave in control of the operation. Following the basic response Substatus, if any, are the Combination Response Extent parameters that belong to the subaddressee(s). If any Substatus is presented, it shall immediately follow the extent to which it applies. The contents of the Major Status and Substatus shall be the same as that defined for all other commands.

5.5.5.1 Slave Address. This field contains the address of the slave in control of the facility to which the response applies.

5.5.5.2 Facility Address. This field contains the address of the facility to which the response applies (x'FF' if slave and facility are integrated).

5.5.5.3 Modifiers. The contents of this octet are echoed by the slave.

5.5.5.4 Residual Count. This field is zero if the extent has been depleted; otherwise, it contains the count of the number of octets or blocks remaining in the extent. This field may have a value even if there were no transfer errors (e.g., a COPY command with a larger Destination extent than Source shall have a Destination extent residual).

5.5.5.5 Data Address. This field contains the address following the last block of transferred data, but may not contain a valid value if the Residual Count is zero.

If the Data Address parameter (see 5.5.11) was used in the parameter list of the command and the Residual Count is nonzero, this field shall be set to x'FFFFFFFF' and the Data Address parameter shall follow the Combination Response Extent parameter.

5.5.5.6 Major Status. This field contains the Response Type and Major Status associated with the subaddressee. The Major Status associated with the basic response packet belongs to the dominant slave.

NOTE: On a Combination command that fails, there may be three sets of Major Status; the dominant slave, the primary subaddressee, and the secondary subaddressee.

5.5.5.7 Substatus. The Substatus, if any, associated with the addressee (primary or secondary) shall be appended and use the same format as that described in 5.4. Extended Substatus, if any, shall also be appended, unless inhibited by Attributes.

5.5.6 Access Key Parameter (Optional) (See Table 18). If the slave provides ACCESS PERMITS to read-protect or write-protect areas of the media, then every operation that accesses such

Table 16
Combination Command Extent Parameter

@	LTH	ID	OCTET	X/b	DEF	COMBINATION COMMAND EXTENT PARAMETER		
M	n+1	33	01			Slave Address		
			02			Facility Address		
			03			MODIFIERS		
				7		0=Primary(Source)	1=Secondary(Destination)	
				6-4		reserved		
				3		Direction	0=Forward	1=Reverse
				2		Addressing	0=DataBlock	1=PhysicalBlock
				1		Data Recovery	0=On	1=Off
				0		Count	0=Octet	1=Block
			04			reserved		
			05-08			Count		
			09-0C			Data Address		
			n-7:4			Count		Repeated as many
			n-3:n			Data Address		times as needed

Table 17
Combination Response Extent Parameter

@	LTH	ID	OCTET	X/b	DEF	COMBINATION RESPONSE EXTENT PARAMETER		
S	n+1	34	01			Slave Address		
			02			Facility Address		
			03			MODIFIERS		
				7		0=Primary(Source)	1=Secondary(Destination)	
				6-4		reserved		
				3		Direction	0=Forward	1=Reverse
				2		0=DataBlock	1=PhysicalBlock	
				1		Data Recovery	0=On	1=Off
				0		Count	0=Octet	1=Block
			04			reserved		
			05-08			Residual Count		
			09-0C			Data Address		
			0D-0E			Major Status		
			0F- n			Substatus if any - Codes x0-xB		

Table 18
Access Key Parameter

@	LTH	ID	OCTET	X/b	DEF	ACCESS KEY PARAMETER
M	05	35	01-04			Access Key

an area shall use the correct access key. (See ACCESS PERMITS command (6.10)).

The Access Key parameter shall precede the Extents parameter to which it refers. If not present, the assumed partition shall be the default data partition.

5.5.7 reserved

5.5.8 reserved

5.5.9 Invalid Parm Parameter (Optional) (See Table 19). This parameter is used by the slave as a response to identify invalid parameters of the command being initiated.

5.5.9.1 Displacement of Parameter in Error. This field is a value that identifies the invalid parameter by its displacement from Octet 0 of the command. A value of x'FFFF' shall be supplied if the slave cannot provide the correct value.

5.5.9.2 Displacement of Field in Error. The second value identifies the offset of the field in error within the parameter in error. The parameter string is the parameter in error beginning at the Length field, up to and including the field in error.

5.5.10 Missing Parm Parameter (Optional) (See Table 20). This parameter is used to identify missing command parameters.

If the slave can determine that it was not given a complete set of parameters, but cannot specifically identify which is missing, the parameter shall consist of the Parameter ID and no ID fields.

5.5.11 Data Address Parameter (Optional) (See Table 21). This parameter is used when the four octets of Data Address in the Extent parameters are insufficient, or if there is a need for the master to use Absolute Addressing. The Absolute Address modifier in the parameter overrides the setting of Bit 2 in the Operation modifier of the command.

For some devices (such as optical disk), the master may use "broad" addressing (such as bands), and the slave needs to advise the master of the exact location at which the data shall be placed. On a WRITE command, the slave may append this parameter to the Transfer Notification packet to inform the master of the actual starting position on the media where the data shall be recorded.

5.5.12 Block Size Parameter (Optional)(See Table 22). This parameter is used when the block size set in the Attributes is to be overridden.

5.5.13 Transfer Parameters (See Table 23). The master may choose from several alternatives in the manner that the slave is to handle a transfer to the recording media. The usage of these alternatives is not typical on every transfer between slave and master, but are a requirement to ensure that certain commands execute in accordance with the master's requirements.

5.5.13.1 Verify. If the Verify modifier is set, the slave shall perform an implied read operation following a write to the media. This may be done to ensure the integrity of the recorded data by using the error detection/error correction code(s) of the device. The verify operation is performed following writing. The facility shall access the block specified by the Data Address. When access is complete, the facility shall verify correct access position and shall read data starting with the addressed block. The slave does not transfer data to the master.

During a multiple-block verify operation, if access boundaries are encountered, the facility shall perform appropriate access movements and verify block positioning. If a defective block for which an alternate location has been assigned is encountered, the facility shall access the alternate block for verifying and then reestablish the access position for subsequent blocks to be read, if any. Access positioning in all cases shall be verified.

Recovery from all data errors detected during verifying shall be attempted by the slave. If the data error cannot be corrected by the slave, processing of the command shall be terminated with a Machine Exception indicated in Major Status. The cause of the termination shall be indicated in Substatus and Extended Substatus (if applicable). The response packet shall identify the block in error. If the data error is corrected, the facility shall continue the verify operation on the next block; the response packet shall indicate that recovery was employed.

NOTE: The slave actions for error correction are vendor specific and may or may not include rewriting the block in error, or automatic relocation of the data to an alternate block in the case of a media defect.

5.5.13.2 Volume. If the Volume modifier is set, the slave shall ignore the Command Extent and associate the requested command with an action beginning at the first block, and completing at the last block on the media.

5.5.13.3 Certify. The Certify modifier applies to the FORMAT command and when set, the slave shall perform whatever actions are necessary to ensure that all recording spaces are free of defects.

Table 19
Invalid Parm Parameter

@	LTH	ID	OCTET	X/b	DEF	INVALID PARM PARAMETER
S	n+1	38	01-02			Displacement of Parameter in error
			03-04			Displacement of field in error
			05- n			Parameter string

Table 20
Missing Parm Parameter

@	LTH	ID	OCTET	X/b	DEF	MISSING PARM PARAMETER
S	n+1	39	01			ID
			n			ID Repeated as many times as needed

Table 21
Data Address Parameter

@	LTH	ID	OCTET	X/b	DEF	DATA ADDRESS PARAMETER
B	n+1	3A	01	7		DataBlock
				6		PhysicalBlock
				5		Absolute Address
				4		Index
				3-0		Vendor Unique
			02			reserved
			03- n			Data Address

Table 22
Block Size Parameter

@	LTH	ID	OCTET	X/b	DEF	BLOCK SIZE PARAMETER
B	05	3B	01-04			Block Size

Table 23
Transfer Parameters

@	LTH	ID	OCTET	X/b	DEF	TRANSFER PARAMETERS
M	02	3C	01	7		Verify
				6		Volume
				5		Certify
				4		Stop on Data Error
				3		
				2		
				1		Compare - use buffer under slave control *
				0		Compare - Master repeat transfer *
						* mutually exclusive parameters

5.5.13.4 Stop on Data Error. If the Stop on Data Error modifier is set, the slave shall stop data transfer immediately following a block on which some error recovery action had been taken. If a multiple block transfer is in progress, the transfer shall be completed as Conditional Success, and a Response Extent is required, which includes the Residual Count. The master shall have to recognize that a Residual Count greater than zero requires the transfer to be re-requested beginning at that point.

5.5.13.5 Compare. If one of the Compare modifiers is set, the data previously written by the slave shall be read back and compared. The source of the data comparison shall be either retransmittal by the master, or from a buffer under the slave's control.

5.5.14 Encapsulation Parameter (Optional) (See Table 24). This parameter may be used by the slave or the master to encapsulate information that cannot be passed directly to or from the addressee, and the contents may or may not be structured as an IPI parameter.

5.5.15 Partition Parameter (Optional) (See Table 25). There may be more than one addressable area on some addressees. This parameter provides for the identification of these areas, and is used whenever the area that a command is to execute upon is other than the default data partition.

5.5.15.1 Disk Partitions. The Partition parameter shall precede the Extents parameter to which it refers. If not present, the assumed partition shall be the default data partition.

The allocation of partitions shall be as follows:

- Slave-Defined
  - Data Partition
    - 00 Default Data Partition
  - Maintenance Partitions
    - 01 CE Partition
    - 02 IML Partition
    - 03-07 Slave-Unique Partitions
- Master-Defined
  - Maintenance Partition
    - 08-0F Additional Maintenance Partitions
  - Data Partitions
    - 10-EF Additional Data Partitions
  - Specific Usage
    - FF IPI usage (e.g., in Attributes, to reference all partitions to any command that accesses partitions 01-0F)

The CE partition defines the area reserved on the device for the exclusive use of diagnostic and other procedures executed by manufacturer's Customer Engineering organization.

The IML partition defines the area reserved on the device for exclusive use by the slave, the facility, or both, to execute an Initial Microprogram Load.

5.5.15.2 Tape Partitions. The Partition parameter shall precede the Extents parameter to which it refers. If not present, the assumed partition shall be the current partition. If present, the Partition parameter shall cause a change to the named partition (if required), followed by positioning. The Partition parameter is not required for subsequent operations once a change has been made. The Partition parameter without an Extent parameter in the OPERATING MODE command shall cause the change to occur at the beginning of a partition.

Tapes retain their position within a partition between commands when the partition ID is x'00' or in the range x'10-FE'.

It is not required that position or other information on Attributes be saved for prior partitions, although some vendors may provide information concerning position retention and other Attributes for prior partitions, or carry over Attributes from the prior partition to the new partition; for example, recording format or DataBlock size (see vendor specification).

The allocation of partitions shall be as follows:

- Group 1
  - Slave-Defined
    - Data Partition
      - 00 Default Data Partition
    - Maintenance Partitions
      - 01 CE Partition on Storage Volume
      - 02 IML Partition
      - 03 CE Partition in Slave or Facility
      - 04-07 Slave-Unique Partitions
  - Master-Defined
    - Maintenance Partitions
      - 08-0F Additional Maintenance Partitions
- Group 2
  - Slave-Defined
    - Data Partition
      - 10-7F Additional Data Partitions (112)
  - Master-Defined
    - Data Partition
      - 80-FE Additional Data Partitions (127)
- Specific Usage
  - FF IPI usage

Table 24
Encapsulation Parameter

@	LTH	ID	OCTET	X/b	DEF	ENCAPSULATION PARAMETER
B	n+1	3D	01- n			Contents to be passed

Table 25
Partition Parameter

@	LTH	ID	OCTET	X/b	DEF	PARTITION PARAMETER
B	n+1	3E	01			Partition ID
			02			reserved
			03- n			Vendor specific

Table 26
Stop on Discontinuity Parameters

@	LTH	ID	OCTET	X/b	DEF	STOP ON DISCONTINUITY PARAMETERS
B	n+1	3F	01			DISCONTINUITY TYPE
				7		Cylinders
				6		Tracks
				5		Access Boundary (set by Access Permits)
				4		Discontiguous Defect Reallocation
				3		Bands
				2		
				to		reserved
			02	0		Time (Discontinuity Value required)
			03-06			Discontinuity Time (in usecs)

The master shall Chain, Sequence, or Order the OPERATING MODES command to any command that accesses a maintenance partition.

The two CE partitions allow for one partition to be on the removable volume and for one to be contained in the addressee. The addressee CE partition shall be contained in semipermanent storage and logically separate from storage used for the Save and Restore of Attributes.

The IML partition provides a capability in the addressee for exclusive use by the slave, the facility, or both, to execute an Initial Microcode Load.

The additional slave-defined data partitions are areas separate from, or in addition to, the default data partition.

The additional master-defined data partitions are allocated from within the default data partition.

5.5.16 Stop On Discontinuity Parameter (Optional) (See Table 26). On some devices, it is desirable that the master be able to identify a discontinuity as a reason to terminate a transfer. This parameter may be associated with the READ, WRITE, and SEARCH transfer commands, or with FORMAT and REPORT DISCONTINUITY. It shall precede the Extent parameter to which it refers.

Table 27
Imbedded Data Parameter

@	LTH	ID	OCTET	X/b	DEF	IMBEDDED DATA PARAMETERS
S	n+1	40	01- n			Data to be transferred

Table 28
Facility Operation Parameters

@	LTH	ID	OCTET	X/b	DEF	FACILITY OPERATION PARAMETERS
B	n+1	3D	01- n			ENCAPSULATION PARAMETER

5.5.16.1 Cylinders. The discontinuity is due to the physical cylinder boundary on the disk.

5.5.16.2 Tracks. The discontinuity is due to the physical track boundary on the disk.

5.5.16.3 Access Boundary. The discontinuity is due to a boundary established by a previously issued ACCESS PERMITS command.

5.5.16.4 Discontiguous Defect Reallocation. The discontinuity is due to the discontiguous reallocation by the slave of a block that included a media defect.

5.5.16.5 Bands. The discontinuity is due to the need for a physical motion on the optical disk, outside the range of a mirror adjustment.

5.5.16.6 Time. This modifier is set to indicate that the time field is to be used.

5.5.16.7 Discontinuity Time. This is a value in microseconds that, if exceeded by the media-accessing techniques, is to be considered a discontinuity.

5.5.17 Imbedded Data Parameter (Optional) (See Table 27). This parameter is used for transferring data that has been gathered by an addressee that has limited, low rates of data transfer requirements.

6. Control Commands

The commands in this section are used as Control commands.

The abbreviations used in graphical representation of commands and responses in the following subsections shall be as follows:

PKT LTH	Packet Length
REF NO	Command Reference Number
OP CODE	Operation Code
COM MOD	Common Command Modifiers
OP MOD	Op Code Modifiers
SLAV ADDR	Slave Address
FAC ADDR	Facility Address
x	Hexadecimal Value
e	Echoed Value (from Command Packet)
b	Bit Value (identified by position 0-7 or 0-B)

6.1 NOP

6.1.1 Command Packet. The command packet for this command shall be as shown in Figure 20.

6.1.2 Response Packet. The response packet for this command shall be as shown in Figure 21.

6.1.3 Description. The NOP command is a null command that is issued to the slave. The slave shall perform no operation, shall not change its state, and shall return the standard completion response to the master. The slave shall clear any previous status (see 5.4) when it executes a NOP command in the same manner as would be done on any other command except REPORT ADDRESSEE STATUS.

The slave shall ignore any parameters associated with this command. Some masters use command chains or sequences that are modified at the time of dispatch to the slave. The NOP opcode may be used to override execution of commands that have associated parameters.

6.2 FACILITY OPERATION

6.2.1 Command Packet. The command packet for this command shall be as shown in Figure 22.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	PARAMETERS 7 through n
xxxx	xxxx	00	bbbb bbbb	xx	xx		

Figure 20
Command Packet for NOP

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 7	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 21
Response Packet for NOP

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	PARAMETERS 7 through n
xxxx	xxxx	01	bbbb bbbb	xx	xx		

Figure 22
Command Packet for FACILITY OPERATION

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 7	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 23
Response Packet for FACILITY OPERATION

6.2.2 Response Packet. The response packet for this command shall be as shown in Figure 23.

6.2.3 Description. The FACILITY OPERATION command is a "pass-through" command that allows a primitive command or a vendor-unique command to be transferred to the slave or facility. The command information is passed in the Encapsulation Parameter appended to the command. The format or content of the encapsulation is slave or facility dependent and is not specified.

6.2.4 Facility Operation Parameter (Encapsulation Parameter). The parameter shall be as shown in Table 28. The contents of this parameter is interpreted directly by the addressee, and may or may not conform to an IPI structure. Unless addressed to the slave, there will usually be no attempt by the slave to parse the encapsulated contents.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR	PARAMETERS 6 through n
xxxx	xxxx	02	bbbb bbbb 7654 3210	xx	xx		

Bits 0, 1, 3 have encoded meaning of:
x'0'=Report
x'1'=Initialize x'2'=Restore
x'9'=Load x'A'=Save

Figure 24
Command Packet for ATTRIBUTES

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 25
Response Packet for ATTRIBUTES

6.3 ATTRIBUTES

6.3.1 Command Packet. The command packet for this command shall be as shown in Figure 24.

6.3.2 Response Packet. The response packet for this command shall be as shown in Figure 25.

6.3.3 Description. The ATTRIBUTES command allows modification of the slave or facility attributes that are used to tell the master what the addressee's operational characteristics are and to allow them to be examined or modified. The operational characteristics that may be modified in the addressee are implementation dependent.

The operating mode of ATTRIBUTES is determined by the opcode modifier, which allows the master to Initialize, Report, Restore, Load, or Save the addressee attributes. The modifiers are mutually exclusive.

NOTE: Bits 0, 1, and 3 (x'1', x'2', and x'8') are encoded.

The information that the slave supplies or modifies, or supplies and modifies, in accordance with the command, is specific to the addressee identified by the Slave Address and Facility Address of the command. Unless a master has prior knowledge, processing of Attributes should begin by analysis of the slave's capabilities and proceed through each attached operational facility.

The opcode modifiers shall be as follows:

(1) Report requires the addressee to respond with a list of parameters that detail the attributes requested by the command.

(2) Initialize allows the master to require the addressee to set all of its attributes to their initial values.

NOTE: These attributes may not represent a valid configuration.

(3) Restore allows the restoration of saved attributes. At power on, slaves and facilities shall perform an automatic Restore. If no attributes have been Saved by the master, the Restore values shall be a valid configuration of the Initialize attributes.

(4) Save allows the addressee attributes, including those associated with this command, to be saved prior to power down or removal of the media from a removable media facility.

(5) Load requires the addressee to modify attributes within the addressee (if they are valid).

When the Initialize or Restore modifiers are set, the addressee acts upon all attributes.

Similarly, if no parameters are transmitted with the command packet when the Report or Save modifiers are set, the addressee acts upon all attributes.

If the master wishes to be selective about attributes to be affected, it shall provide a list of the parameter IDs (via the Request Parm parameter), with the Report, Load, or Save modifiers.

The master and slave either have parameters that are unique to each (so indicated by either M or S in the @ column), or are common to both (indicated by "B" in the @ column). Common parameters are used by the slave to report, and by the master to modify. A consistent sequence is necessary to properly manage parameters that are common.

If a master wishes to find out the Initial settings of the slave (rather than the Restored settings), it issues an ATTRIBUTES command with the Initialize modifier set. The slave shall set the Attributes parameters to their initial factory values. The master issues an ATTRIBUTES command with the Report modifier set, to look at the parameter or parameters of interest.

The master can change the Attribute parameters by issuing an ATTRIBUTES command with the Load or Save modifier set, and thus instruct the slave to act upon the new values.

If the master does not wish the new values to be kept beyond Power Off, the Load modifier is set.

If the master wishes the new values to be kept beyond Power Off, and Restored by the slave after Power On, the Save modifier is set.

The master can use the Restore modifier to have the slave return to its previously Saved values.

Some of the attributes apply equally to either slave or facility (e.g., number of ports). In the case of an integrated slave and facility, both slave and facility apply. For this reason it is impossible to clearly define attributes as belonging to either slave or facility unless the configuration of intended use is known. Therefore, all attributes are shown as being relative to the addressee, even though some may be specifically slave oriented, and others may be specifically facility oriented.

Within the parameters there are sets of octets that may need to be repeated several times to provide all of the information. These repetitive octet sets are noted in the parameter tables.

On facilities that support more than one type of partition, the Partition Parameter shall precede every set of attribute parameters for that partition. In this manner, every partition is described by a group of succeeding attribute parameters (e.g., if a disk that has been formatted with one PhysicalBlock size has three partitions, the Size of disk PhysicalBlocks parameter would be the same and repeated in every set of parameters succeeding each Partition parameter.

If the Report modifier is set, and a Partition Parameter with an ID of x'FF' is appended to the command, the slave shall respond with information on all of the partitions (with each set of information preceded by a Partition Parameter). On a Report, Load or Save, the absence of a Partition Parameter means the default data partition attributes are to be referenced.

If any fields are not needed in a parameter, the parameter length can be cut short (e.g., on Parameters 53 and 54, a disk with a fixed clock rate and variable rotation speed has a different number of bytes per track on every cylinder so parameters such as the Total Number of Blocks per Cylinder and Total Number of Blocks per Track need not be supplied.

The following modifiers permit operations upon individual Attributes:

(1) *Report.* The current memory contents are reported to the master. If no parameters are appended, the slave responds with all attributes (which can be a very large length). The Request Parms parameter may be used to specifically identify Attributes.

(2) *Load.* This modifier requires that parameters be appended for Attributes that may be modified. The slave shall replace the contents of the designated parameters in Current memory with the ones in the command parameter list (if valid).

(3) *Save.* If this modifier has associated parameters, the command is executed in the same manner as a Load, then the contents of Current memory shall be written into Semi-Permanent memory.

The following modifiers are those that operate upon all changeable Attributes:

(1) *Initialize.* No parameters are accepted. The contents of Permanent memory shall be written into Current memory.

(2) *Restore.* No parameters are accepted. The contents of Semi-Permanent memory are written into the Current memory.

(3) *Save.* If no parameters are appended, the contents of Current memory are written into Semi-Permanent memory.

Implementation Note: There is a need for three types of memory to completely manage Attributes. To retain Attributes, permanent memory shall be used; to retain Attributes changed by the master, semi-permanent memory shall be used.

(1) *Permanent.* This memory contains all of the attributes as defined by the manufacturer and the Initial value of Attributes. The Initial Attributes may not be set to a valid configuration (e.g., two features that are mutually self-exclusive may be capable of being supported by the slave).

(2) *Semi-Permanent.* At the point of manufacture, these values are set to a valid combination of the Initial Attributes. The contents may be replaced by the master performing a Save. The slave uses the contents of this memory to Restore Attributes at Power On, or under command of the master when the Restore modifier is set.

(3) *Current.* After Power On, the contents are the same as Semi-Permanent memory, that is, Restored. Individual Attributes may be changed by the master performing either a Load, or a Save with parameters.

6.3.4 Attribute Parameters

6.3.4.1 Parameters 3A, 3E, 50. These parameters shall be as shown in Table 29.

6.3.4.1.1 Data Address (Common) Parameter. This parameter is used to follow any parameter in which a 32-bit Data Address field is inadequate.

6.3.4.1.2 Partition (Common) Parameter This parameter is used to precede the set or sets of facility-dependent parameters to identify which partition is being referred to.

6.3.4.1.3 Vendor ID Parameter. This parameter is made up of the following:

(1) *Manufacturer Identification (ASCII).* This field contains the Vendor ID of the addressee.

(2) *Manufacturer Model Number (ASCII).* This field contains the model number of the addressee.

(3) *Manufacturer Revision Number (ASCII).* This field contains the current revision number of the addressee.

(4) *Manufacturer Unique ID.* This field contains a value that is a unique identification of the slave. If the slave manufacturer does not provide a unique value, then this field shall be modifiable by the master so as to provide a unique ID value.

(5) *Manufacturer Switch Settings.* This field contains the settings of switches that may be field set or modified on the slave.

(6) *Manufacturer-Defined Fields.* Any fields that the manufacturer chooses to provide (vendor defined).

6.3.4.2 Parameters 51-58. These parameters shall be as shown in Table 30.

6.3.4.2.1 Size of Disk DataBlocks Parameter. This field contains an unsigned binary number specifying the size of the DataBlocks contained within the disk Partition. The master may use this parameter to set the DataBlock size of a formatted slave.

6.3.4.2.2 Size of Disk PhysicalBlocks Parameter. This field contains an unsigned binary number specifying the size of the PhysicalBlock contained within the disk Partition. PhysicalBlock size is established by the FORMAT command.

6.3.4.2.3 Total Number of Disk DataBlocks Parameter

(1) *Total Number of Blocks per Partition.* An unsigned binary number specifying the number of DataBlocks contained within the disk partition.

(2) *Total Number of Blocks per Cylinder.* An unsigned binary number specifying the number of DataBlocks per cylinder.

(3) *Total Number of Blocks per Track.* An unsigned binary number specifying the number of DataBlocks per track.

(4) *Data Address.* This field contains the starting address of the first block. The first data address shall always be zero. On disks that have a variable number of octets per track, typically within bands or groups of cylinders, to define a partition requires the use of more than one set of fields, so they are repeated as many times as necessary.

6.3.4.2.4 Total Number of Disk PhysicalBlocks Parameter. This parameter follows the same format as that for DataBlocks, except that the field contents refer to PhysicalBlocks and not DataBlocks.

6.3.4.2.5 DataBlock Sizes Supported Parameter

(1) *Smallest Block Size Supported.* An unsigned binary number specifying the smallest value of a range of DataBlock sizes.

(2) *Largest Block Size Supported.* An unsigned binary number specifying the largest value of the range supported.

(3) *Increment Size.* An unsigned binary number specifying the increment by which a block can increase from the smallest to the largest size.

Table 29
Attribute Parameters 3A, 3E, 50

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
B	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
B	n+1	50				VENDOR ID
			01-10			Manufacturer Identification (ASCII)
			11-18			Manufacturer Model Number (ASCII)
			19-1C			Manufacturer Revision Number (ASCII)
			1D-24			Manufacturer Unique ID
			25-28			Manufacturer Switch Settings
			29- n			Manufacturer-Defined Fields

Table 30
Attribute Parameters 51-58

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS	
B	05	51	01-04			SIZE OF Disk DataBlocks *	
S	05	52	01-04			SIZE OF Disk PhysicalBlocks *	
						* If zero, the addressee is not formatted and/or IDs 53 and 54 are invalid	
S	n+1	53				TOTAL NUMBER OF Disk DataBlocks	
			01-04			Total No of Blocks per Partition	
			05-08			Total No of Blocks per Cylinder	
			09-0C			Total No of Blocks per Track	
			0D-10			Data Address	
			n-B:8			Total No of Blocks per Cylinder	repeated as
			n-7:4			Total No of Blocks per Track	many times
			n-3:n			Data Address	as needed
S	n+1	54				TOTAL NUMBER OF Disk PhysicalBlocks	
			01-04			Total Number of Blocks per Partition	
			05-08			Total Number of Blocks per Cylinder	
			09-0C			Total Number of Blocks per Track	
			0D-10			Data Address	
			n-B:8			Total No of Blocks per Cylinder	repeated as
			n-7:4			Total No of Blocks per Track	many times
			n-3:n			Data Address	as needed
S	n+1	55				DataBlock SIZES SUPPORTED	
			01-04			Smallest Block Size Supported	
			05-08			Largest Block Size Supported	
			09-0C			Increment Size	
			n-B:8			Smallest Block Size	repeated as
			n-7:4			Largest Block Size	many times
			n-3:n			Increment Size	as needed
S	n+1	56				PhysicalBlock SIZES SUPPORTED	
			01-04			Smallest Block Size Supported	
			05-08			Largest Block Size Supported	
			09-0C			Increment Size	
			n-B:8			Smallest Block Size	repeated as
			n-7:4			Largest Block Size	many times
			n-3:n			Increment Size	as needed
S	n+1	57				SIZE OF PHYSICAL GROUPS	
			01-02			Number of PhysicalBlocks per Physical Group	
		58				reserved	

There may be more than one range supported, so these fields are repeated as many times as required for the disk. If a block size is fixed, the same value is duplicated in the Smallest and Largest fields.

If a disk can be formatted with any block size greater than 256 up to the end of track with 20,160 octets, it would be represented by one set - (256,20160,1). If only two multiples of block size, 512 and 2048, are supported per track two sets would be represented (e.g., (512,17408,512) followed by (2048,18432,2048)).

6.3.4.2.6 PhysicalBlock Sizes Supported Parameter. This parameter follows the same format as that for DataBlocks, except that the field contents refer to PhysicalBlocks and not DataBlocks.

6.3.4.2.7 Size of Physical Groups. This parameter contains an unsigned binary value with the count of the number of PhysicalBlocks in a Physical Group.

6.3.4.3 Parameters 59-5A. These parameters shall be as shown in Table 31.

6.3.4.3.1 Attribute Table Conditions Parameter. The following Attribute table conditions may be listed in this parameter:

(1) *Attribute Table May be Corrupted.* If a command to change Attributes is initiated, this bit is set and all others are set to zero. If the command fails to complete successfully, then this bit shall remain set to indicate that there may have been contamination by partial processing. If the command completes successfully, then this bit is reset to zero and the appropriate bit setting listed in (2) through (6) is posted.

(2) *Attribute Table Initialized by Master.* This bit is set by the slave to indicate that the attributes have been set to their initial values by command of the master. If the master attempts to change any attributes after this bit has been set, the slave shall reset this bit to indicate that attributes have been changed.

(3) *Attribute Table Restored by Slave.* This bit is set by the slave following Power On to indicate that the attributes have been restored. If the master attempts to change any attributes after this bit has been set, the slave shall reset this bit to indicate that attributes have been changed.

(4) *Attribute Table Restored by Master.* This bit is set by the slave to indicate that the attributes have been set to their saved values by command of the master. If the master attempts to change any attributes after this bit has been set, the slave shall reset this bit to indicate that attributes have been changed.

(5) *Attribute Table Loaded by Master.* This bit is set by the slave to indicate that attributes have been loaded by command of the master. If the master attempts to change any attributes after this bit has been set, the slave shall reset this bit to indicate that attributes have been changed.

(6) *Attribute Table Saved by Master.* This bit is set by the slave to indicate that attributes have been saved by command of the master. If the master attempts to change any attributes after this bit has been set, the slave shall reset this bit to indicate that attributes have been changed.

6.3.4.3.2 Pad with Fill Characters Parameter. When a situation exists where the slave has to pad data, the slave shall use this field. In the event that the field specified does not fill the entire space to be padded, the slave shall repeat the supplied field until the space is filled.

6.3.4.4 Parameters 5B-5D. These parameters shall be as shown in Table 32.

6.3.4.4.1 Disk Partition Definition Parameter. The Partition ID identifies the Partition to be defined. The Facility Address (it may be an Actual or Synonym address) defines where the partition is located. The Type of disk is bit significant and identifies the kind of disk area in which the partition is located. The Block Count and Data Address are used to define the extent of the formatted default data area that is to be the partition. This parameter can only be issued to a formatted facility.

6.3.4.4.2 Synonym Definition Parameter. The parameter list consists of ordered pairs in which the first octet in each pair contains the Synonym address, followed by the absolute Facility Address. Commands issued by the master can use the synonym in the Facility Address field, and the slave shall be responsible to correctly address the actual facility.

6.3.4.4.3 Alias Definition Parameter. The parameter list consists of three ordered octets in which the first octet contains the Alias address, followed by the Facility Address (which may be a Synonym or an Actual address) and the Partition ID to which it is to refer. The slave shall accept the Alias as a Facility Address to refer to the partition, in lieu of the Partition ID preceding Extent parameters.

6.3.4.5 Parameters 5E-5F. These parameters shall be as shown in Table 33.

Table 31
Attribute Parameters 59-5A

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
S	02	59				ATTRIBUTE TABLE CONDITIONS
			01	7		Attribute Table May be Corrupted
				6		Attribute Table Initialized by Master
				5		Attribute Table Restored by Slave
				4		Attribute Table Restored by Master
				3		Attribute Table Loaded by Master
				2		Attribute Table Saved by Master
			0-1			reserved
B	n+1	5A				PAD WITH FILL CHARACTERS
			01- n			Fill character(s)

Table 32
Attribute Parameters 5B-5D

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS	
B	n+1	5B				DISK PARTITION DEFINITION	
			01			Partition ID	
			02			Facility Address	
			03			Type of Disk	
				7		Nonremovable Disk	
				6		Removable Disk	
				5		Floppy Disk	
				4		Fixed Head Disk	
				3		Moving Head Disk	
				2		Solid State Disk	
				1			
			to			reserved	
			04	0			
			05-08			Block Count	
			09-0C			DataBlock Address	
B	n+1	5C				SYNONYM DEFINITION	
			01			Synonym Address	
			02			Actual Facility Address	
			n- 1			Synonym Address	repeated as many
			n			Actual Facility Address	times as needed
B	n+1	5D				ALIAS DEFINITION	
			01			Alias Address	
			02			Facility Address	
			03			Partition ID	
			04			reserved	
			n- 3			Alias Address	repeated
			n- 2			Facility Address	as many
			n- 1			Partition ID	times as
			n			reserved	needed

Table 33
Attribute Parameters 5E-5F

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
B	n+1	5E				MULTI-PORT CHARACTERISTICS
			01-02			Number of Ports
			03			No of stacked cmds allowed at alt'nate port(s)
B	n+1	5F				PHYSICAL disk CONFIGURATION
			01-04			Address of Last Data Cylinder
			05-08			Address of Defect List Cylinder
			09-0A			Number of Heads per Cylinder
			0B-0E			Number of Fixed Sectors per Revolution *
			0F-12			Absolute Number of Octets per Track
			13-16			Single Cylinder Seek Time (usecs)
			17-1A			Average Cylinder Seek time (usecs)
			1B-1E			Maximum Cylinder Seek time (usecs)
			1F-22			Rotation Time (usecs)
			23-26			Head Switching Time (usecs)
			27-2A			Write-to-Read Recovery Time (usecs)
			2B-n			Vendor Specific Data

* Zero if not fixed.

6.3.4.5.1 Multi-Port Characteristics Parameter. The first field contains the number of ports on the addressee. The second field contains an unsigned binary value of the number of commands that can be stacked at an alternate port before the addressee's alternate port shall report a Busy condition or Command Reject (see 4.10.5) due to the command buffer being full.

If the master wishes to force a single switching point at the addressee, it may force the number of commands allowed at alternate ports to zero.

6.3.4.5.2 Physical disk Configuration Parameter. This parameter may be used by the master to identify the characteristics of disk drives that are not self-identifying to the slave. It is more typically used by the slave to inform the master of the physical characteristics of the disk or disks attached.

(1) *Address of Last Data Cylinder.* The assumed starting data cylinder is zero. Thus this value, +1, specifies the number of data cylinders (including spares) on the addressee.

(2) *Address of Defect List Cylinder.* This value specifies the address of the defect list data cylinder.

(3) *Number of Heads per Cylinder.* This value specifies the number of addressable heads per cylinder for the addressee.

(4) *Number of Fixed Sectors per Revolution.* This value specifies the number of fixed sectors (including spares) available on each revolution for the addressee. Note that sectors between imbedded servo bursts can be configured to contain more than one PhysicalBlock. This value is zero if the disk does not have fixed sectors.

(5) *Absolute Number of Octets per Track.* This value specifies the number of octets nominally included on each track of the addressee; including gaps, spares, and other overhead allowances.

(6) *Single Cylinder Seek Time.* This value specifies the single cylinder seek time for moving head disk drives.

(7) *Average Cylinder Seek Time.* This value specifies the average cylinder seek time for moving head disk drives, as specified by the vendor.

(8) *Maximum Cylinder Seek Time.* This value specifies the maximum cylinder seek time for moving head disk drives.

(9) *Head Switch Time.* This value specifies the head switch time for multiple head disk drives.

(10) *Rotational Period.* This value specifies the rotational period for the addressee (i.e., the time it takes for one disk revolution from index to index).

(11) *Write to Read Recovery Time.* This value specifies the time that it takes the disk to become capable of reading data after writing data.

6.3.4.6 Parameters 60-63. These parameters shall be as shown in Table 34.

6.3.4.6.1 DataBlock Interleave Parameter. The first octet is used to identify whether the interleave is established by values unique to the facility, or by factors that are predictable by the master.

When Bit 7 = 0, the parameter refers to factors that affect transfer rate in a manner that is predictable based upon an algorithm of block interleave factors.

(1) *Supported.* This value specifies the maximum interleave factor that can be supported.

(2) *Current.* This value specifies the current factor of interleave on the formatted disk. A value of 0 or 1 means that there is no interleave factor (i.e., 1:1).

When Bit 7 = 1, the parameter refers to values that are device specific, and cannot be calculated without specific knowledge of a preexisting disk format.

(1) *Supported.* This octet identifies the interleave factor values that are defined by the facility. Bit 0 specifies interleave value 0, which is the basic transfer rate capability of the facility, and is always indicated. Bits 1-7 specify interleave values that are defined by vendor specifications.

7	Lowest Effective Transfer Rate
1-6	Intermediate Effective Transfer Rates
0	Fastest Effective Transfer Rate

Individual bits 1-7 indicate the ability to cause a reduction in the transfer rate that exceeds the reduction of the immediately preceding bit. (When Bit 5 = 1, it will cause a greater reduction in the transfer rate than when Bit 4 = 1.)

(2) *Current.* This octet identifies the interleave factor value that is presently established for the facility. Only one bit shall be set in this octet.

6.3.4.6.2 Transfer Rate Parameter

(1) *Effective Transfer Rate.* This value specifies the effective transfer rate of the addressee (e.g., based on the interleave factors used by the slave at formatting). The value is

Table 34
Attribute Parameters 60-63

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS	
S	n+1	60				DataBlock INTERLEAVE PARAMETER	
			01	7		Interleave Factor/Interleave Values	
				6-0	0		
			02			Supported	
			03			Current	
S	09	61				TRANSFER RATE (Octets/second)	
			01-04			Effective Transfer Rate	
			05-08			Slave's Instantaneous Transfer Rate	
S	n+1	62				PhysicalBlock PERFORMANCE CHAR'STICS SUPPORTED	
			01			Smallest PhysicalBlock Interleave Factor	
			02			Largest PhysicalBlock Interleave Factor	
			03			Smallest Head Interleave Factor	
			04			Largest Head Interleave Factor	
			05			Smallest Cylinder Interleave Factor	
			06			Largest Cylinder Interleave Factor	
			n-5:4			PhysicalBlock Interleave Factor	repeated as
			n-3:2			Head Interleave Factor	many times
			n-1:n			Cylinder Interleave Factor	as needed
S	n+1	63				CURRENT PhysicalBlock PERFORMANCE SETTINGS	
			01			PhysicalBlock Interleave factor	
			02			Head Interleave Factor	
			03			Cylinder Interleave factor	

vendor defined and typically represents a calculation based on the time it takes to transfer one track of data in contiguous block number sequence, which, with an interleave factor of greater than 0, requires more than one rotation.

(2) *Slave's Instantaneous Transfer Rate*. This value specifies the maximum speed at which the slave can transfer from its internal buffer(s).

6.3.4.6.3 PhysicalBlock Performance Characteristics Supported Parameter. This parameter is provided by the slave to advise the master which interleave factors can be supported when PhysicalBlock formatting is used. The three pairs of octets are repeated if there is more than one range that can be supported.

(1) *PhysicalBlock Interleave Factor*. This pair of octets defines the low-to-high range of interleave factors that can be supported between PhysicalBlocks.

(2) *Head Interleave Factor*. This pair of octets defines the low-to-high range of interleave factors that can be supported between heads.

(3) *Cylinder Interleave Factor*. This pair of octets defines the low-to-high range of interleave factors that can be supported between cylinders.

If only one factor can be supported, the same value is repeated in the smallest and largest field.

6.3.4.6.4 Current PhysicalBlock Performance Settings Parameter. This parameter is used to advise the master of the current interleave settings of the partition referred to. If no Partition parameter precedes this parameter, the values for the default data partition are provided.

(1) *PhysicalBlock Interleave Factor*. This octet defines the current interleave factor in effect between PhysicalBlocks.

(2) *Head Interleave Factor*. This octet defines the current interleave factor in effect between heads.

(3) *Cylinder Interleave Factor*. This octet defines the current interleave factor in effect between cylinders.

Table 35
Attribute Parameters 64-65

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
S	15	64				PHYSICAL INTERFACE ATTRIBUTES PARAMETER
			01-04			SDE (nsecs)
			05-08			IRT (nsecs)
			09-0C			CCD (nsecs)
			0D-10			SDR (usecs)
			11-14			SYD (usecs)
S	n+1	65				ADDRESSEE CONFIGURATION PARAMETER
			01-04			Data Buffer Size
			05-08			Command buffer size
			09-0A			Max No of octets in Command Packet
			0B-0C			Max No of octets in Response Packet
			0D			Max No of Access Permit Extents
			0E			Min No of Queued Commands
			0F			Max No of Queued Commands
			10			Size of Command Stack

6.3.4.7 Parameters 64-65. These parameters shall be as shown in Table 35.

6.3.4.7.1 Physical Interface Attributes Parameter. See Section 5 of ANSI X3.129-1986 for the definition of these fields, which must be supplied at Power On.

6.3.4.7.2 Addressee Configuration Parameter. This parameter is typically relevant only to the slave. However, a slave may have facilities attached that contain their own buffers and capabilities, or the slave may have allocated its own buffering on a dedicated basis amongst the facilities. A simple example would be facilities with integral data buffers for error correction (in which case only the first parameter field is required).

This parameter is intended to provide the master with information that would allow it to maximize performance. If this parameter is relevant to facilities as well as the slave, it is noted in the Slave Configuration parameter (see 6.3.4.8(12)).

This parameter is made up of the following parameters:

(1) *Data Buffer Size.* This value specifies the size (in octets) of the data buffer associated with the addressee.

(2) *Command Buffer Size.* This value specifies the size (in octets) of the command buffer associated with the addressee.

(3) *Maximum Number of Octets in Command Packet.* This value specifies the maximum number of octets the addressee can accept in a command packet.

(4) *Maximum Number of Octets in Response Packet.* This value specifies the maximum number of octets the addressee can provide in a response packet.

(5) *Maximum Number of Access Permit Extents per Addressee.* This value specifies the maximum number of Access Permits Extents allowed per addressee.

(6) *Minimum Number of Queued Commands.* Any value greater than 0 indicated the minimum number that shall be guaranteed to queue per facility. A value of zero means the slave cannot guarantee any commands for the facility.

(7) *Maximum Number of Queued Commands.* A value of 0 means that the queue may contain more than the minimum but only as many as the size of the command buffer permits. A value of 1 means that there is no Command Queuing and commands are Individual. Any other value shall be greater than or equal to the minimum.

(8) *Size of Command Stack.* This value specifies the minimum number of commands that the slave is capable of stacking (e.g., a slave with addressability to 8 facilities and a minimum queue size of 4 per facility may stack 32. If this is an absolute limitation, the value of 32 shall be specified. However, in some implementations, the stack may be as large as command buffering and the queuing algorithm permit, and may exceed 32 if the commands are small in length. If the size of the stack is variable, depending on the sum of command sizes, and it can exceed the calculatable maximum, this field shall be set to x'FF'.

Table 36
Attribute Parameter 66

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
S	n+1	66				SLAVE CONFIGURATION (BIT SIGNIFICANT)
			01	7		Facilities may be of Different Classes
				6		Facility-Facility Transfers
				5		Synonym Addressing
				4		Alias Addressing
				3		Odd Octet Transfers
				2		Master Termination of Commands Required
				1		Extended Substatus
				0		Multiplexed Data Transfers
			02	7		Transfer Notification Packets
				6		Imbedded Data Responses
				5		Master-Definable Maintenance Partitions
				4		Facility Configuration Information
				3		Master Throttling of Data Streaming
				2		Multiple Command Extents Accepted
				1		Data Streaming Data Transfers
				0		Interlock Data Transfers
			03	7-0		reserved
			04			Logical Interface
				7		Level 2
				6		Level 3
				5-0		reserved

Implementation Note: A large number of factors influence the exact degree of stacking and queuing actually available in a slave (e.g., a slave capable of addressing 8 facilities but only 3 are physically attached and operational). If Individual commands are issued, then no more than 3 commands can be stacked. If Queued commands are issued, and there may be up to 4 per facility, the command stack would be 32. However, since only 3 facilities are operational, the effective queue per facility would average above 10 commands. The minimum parameters are what the slave guarantees, and are not intended to be an upper limit.

Another consideration is that commands are variable in length, but the size of the command buffer is typically fixed. In a configuration of 8 facilities with a minimum command queue of 4 and a maximum command size of 256, the command stack would be 32 and the command buffer size would have to be 8192 octets (32*256). However, if only one command of maximum size can be accepted by the slave (e.g., a COPY), and the others are typically less than 64 octets, then the command buffer size need be only 2240 octets (256+(31*64). Refer to vendor documentation to determine the methods of buffer management used to maximize performance.

6.3.4.8 Attribute Parameter 66 (Slave Configuration (Bit Significant) Parameter). This parameter shall be as shown in Table 36. There are a large number of features that the slave may be able to support, and the following list provides a summary of its capabilities.

(1) *The Facilities May Be of Different Classes Bit.* This bit shall be set by the slave if it can support more than one class of facility.

NOTE: This bit is set even if both classes are not currently attached.

(2) *Facility — Facility Transfer Capability Bit.* This bit shall be set by the slave to indicate that it has the capability to transfer data between facilities attached to it without master intervention.

(3) *Synonym Address Bit.* This bit shall be set if the slave allows the master to redefine Facility Addresses as Synonyms.

(4) *Alias Address Bit.* This bit shall be set if the slave allows the master to redefine Facility Addresses to refer to data partitions by Aliases.

(5) *Odd Octets Transfer Bit.* This bit shall be set if the slave supports Double Octet Mode and can transfer odd octets of information.

(6) *Master Termination of Commands Required Bit.* This bit shall be set if the slave requires the master to terminate the Information Transfer of commands (i.e., the slave does not dynamically use the Packet Length of the command to generate the necessary number of SYNC INs at the Physical Interface).

Table 37
Attribute Parameter 67

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
S	n+1	67				SLAVE CONFIGURATION (FIELDS)
			01			No of Synonyms Supported per Actual Address
			02			Lowest Synonym Address Supported
			03			Highest Synonym Address Supported
			04			No of Data Partitions Supported per facility
			05			No of Aliases Supported per Partition
			06			Lowest Alias Address Supported
			07			Highest Alias Address Supported
			08			No of concurrent COPY commands supported
			09			Maximimum No of extents Supported
			0A-0F			reserved
			10- n			Actual Addresses of Facilities Attached

(7) *Extended Substatus Bit.* This bit shall be set if the slave can provide Extended Substatus. If this capability is available, the slave shall also be capable of inhibiting Extended Substatus in responses.

(8) *Multiplexed Data Transfers Bit.* This bit shall be set if the slave is capable of multiplexing data transfers (i.e., breaking up a single transfer request into bursts defined as acceptable by the master).

(9) *Transfer Notification Packets Bit.* This bit shall be set if the slave has the capability to generate these packets.

(10) *Imbedded Data Responses Bit.* This bit shall be set if the slave has the capability to generate these responses.

(11) *Master-Definable Maintenance Partitions Bit.* This bit shall be set if the slave is capable of allowing the master to define a maintenance partition within the default data partition.

(12) *Facility Configuration Information Bit.* This bit shall be set if the slave can provide Addressee Configuration parameters for attached facilities.

(13) *Master Throttling of Data Streaming Bit.* This bit shall be set if the slave can support the master throttling a data streaming transfer.

(14) *Multiple Command Extents Accepted Bit.* This bit shall be set if the slave is capable of accepting more than one Command Extent on a READ or WRITE command, in order to permit "scatter READ" and "scatter WRITE."

(15) *Data Streaming Data Transfers Bit.* This bit shall be set if the slave is capable of Data Streaming data transfers.

NOTE: In this case, the Physical Interface defines operation transfers.

(16) *Interlocked Data Transfers Bit.* This bit shall be set if the slave is capable of interlocking data transfers.

NOTE: In this case, the Physical Interface defines operation transfers.

(17) *Level 2 Bit.* This bit shall be set if the slave can support Level 2 operations.

(18) *Level 3 Bit.* This bit shall always be set.

6.3.4.9 Parameter 67 (Slave Configuration (Fields) Parameter). This parameter shall be as shown in Table 37 and is made up of the following octets.

(1) *Number of Synonyms Supported per Actual Address.* This value specifies the number of Synonym addresses that can be used to refer to any one Actual address. If the slave cannot support Synonym addressing, this value is 0, and the contents of the next two octets shall be x'FF'.

(2) *Lowest Synonym Address Supported.* This value defines the lowest Synonym address supported as a Facility Address (typically 00).

(3) *Highest Synonym Address Supported.* This value defines the highest Synonym address supported as a Facility Address (may be less than or equal to FE).

Table 38
Attribute Parameter 68

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS	
B	n+1	68				FACILITIES ATTACHED TO SLAVE PARAMETER	
			01			Actual Facility Address	
			02			Facility Class	
				01		Magnetic Disk	
				02		Optical Disk	
				03		Magnetic Tape	
				04		Communications	
			03			Facility Type	
						For Magnetic Disk:	
				7		Nonremovable Disk	
				6		Removable Disk	
				5		Floppy Disk	
				4		Fixed Head Disk	
				3		Moving Head Disk	
				2		Solid State Disk	
				1			
				to		reserved	
			04	0			
			n- 3			Actual Facility Address	repeated
			n- 2			Facility Class	as many
			n- 1			Facility Type	times as
			n			reserved	needed

(4) *Number of Data Partitions Supported per Facility.* This value specifies the number of partitions that can be supported in the data area of a facility. A value of 1 means that the slave supports only the default data partition.

NOTE: If the slave also supports Alias Addressing, then a partition may be referred to by an Alias.

(5) *Number of Aliases Supported per Partition.* This value specifies the number of Alias addresses that can be used to refer to a Partition. If the slave cannot support Alias addressing, this value is 0, and the contents of the next two octets shall be x'FF'.

(6) *Lowest Alias Address Supported.* This value defines the lowest Alias address supported as a Facility Address (typically 10).

(7) *Highest Alias Address Supported.* This value defines the highest Alias address supported as a Facility Address (may be less than or equal to FE).

(8) *Number of Concurrent COPY Commands Supported.* This value specifies the number of COPY commands that can be outstanding for the slave.

(9) *Maximum Number of Extents Permitted.* This value specifies the maximum number of extents that can be specified on those commands that allow multiple extents to be supplied. If a value of 0 is set, it means that the number is as large as can be incorporated in the slave's command buffer; this will vary depending on the size of the command buffer and the number of other commands already residing in the buffer.

(10) *Actual Addresses of Facilities Attached.* This is a string of octets, each of which identifies an attached facility by its Actual Address. The address assignment is not necessarily contiguous for facilities. All facilities attached shall be identified.

NOTE: Information on facilities that are not Powered-Up or are otherwise nonfunctional may not be available via the Facilities Attached parameter.

6.3.4.10 Parameter 68 (Facilities Attached to Slave Parameter). This parameter shall be as shown in Table 38. This parameter is addressed only to the slave, and its contents are repeated for as many facilities as there are attached to the slave. It would typically be used by the slave to advise the master of attached facilities. However, when facilities attached are not self-identifying, the master may use this parameter to advise the slave of the attached facility characteristics.

The first octet contains the Actual address of the facility that is typically a device, and the next pair is used to identify the type of facility. The first octet of the pair defines the generic class of device, and the second octet is bit significant to allow reporting of devices that have more than one characteristic (e.g., a disk may have both fixed and moving heads).

Table 39
Attribute Parameters 69-6A

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
		69				reserved
S	n+1	6A				COMMAND SUPPORTED
			01			Op Code Supported
			02			reserved
			03-04			Common Modifier Mask
			03			reserved
			04	F-8		reserved
				7		Priority/Ordered
				6		Priority/Sequential
				5		Priority/Chained
				4		Priority
				3		Ordered
				2		Sequential
				1		Chained
				0		Queued
			05-06			Op Code Modifier Mask
				F-0		Coded According to Modifiers Supported
			07			Parameter ID (first) repeated as many
			n			Parameter ID (last) times as needed

6.3.4.11 Parameters 69-6A

6.3.4.11.1 Parameter 69. Reserved.

6.3.4.11.2 Command Supported Parameter). This parameter shall be as shown in Table 39. This parameter is present for every command supported by the slave and is made up of the following octets.

(1) *Op Code Supported.* The value of the supported Op Code is supplied.

(2) *Common Modifier Mask.* Each bit position represents the encode value of the common modifier bits implemented by the slave for the command (e.g., if a slave does not support Sequential or Ordered on a command, then bit positions 1, 4, and 5 are set to 1 to indicate that Priority and/or Chained are supported in any combination).

(3) *Op Code Modifier Mask.* Each bit represents the encode value of the Op Code Modifier bits implemented by the slave on a command.

(4) *Parameter ID.* This is a list of the IDs of all of the parameters that are supported for the specified op code.

6.3.4.12 Parameter 6B (Masks of Octets Supported Parameter). This parameter shall be as shown in Table 40 and are made up of the following octets.

(1) *Selection Octet Mask.* Bits 4-6 are required for execution over the Physical Interface, so the corresponding bits shall be set to 1. If any of the Bits 7 or 0-3 are set to 1, it indicates that the slave supports those capabilities at the Physical Interface (e.g., if Bit 7 is set to one, the slave supports the Facility Selection option).

(2) *Bus Control Octet Mask.* Bits 6-7 are required for execution over the Physical Interface so the corresponding bits shall be set to 1. If any of the Bits 4-5 are set to 1, it indicates that the slave supports those capabilities at the Physical Interface. The Generic command set reserves these bits.

(3) *Bus Acknowledge Octet Mask.* Support of this octet is optional at the Physical Interface and any bits set to 1 shall indicate support of this octet.

(4) *Master Status Octet Mask.* The setting of any of the Bits 4-6 indicates that the slave supports the options at the Physical Interface (e.g., the Slave — Slave Operation Completed Bit is set by the slave to indicate that it has the capability to perform Slave — Slave Information Transfers (see also ANSI X3.129-1986). Support of defined Ending Status is required, and no other use of these bits may be made.

NOTE: If no Ending Status codes can be provided, "0000" shall be reported in the octet during the Ending Status sequence (see 4.10.4.1).

Table 40
Attribute Parameter 6B

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
S	n+1	6B				MASKS OF OCTETS SUPPORTED
			01			Selection Octet Mask
				7		1=Facility Selection supported
				4-6	7	Slave Address - Mandatory
				3		1=Change Transfer Mode
				2		1=Change Octet Mode
				1		1=Priority Select
				0		1=Priority Hold
			02			Bus Control Octet Mask
				7	1	Operation/Data - Mandatory
				6	1	Direction - Mandatory
				5		1=reserved
				4		1=Control of Bus
				0-3		1-F=reserved
			03			Bus Acknowledge Octet Mask
				7		1=Operation/Data
				6		1=Transfer Direction
				5		1=reserved
				4		1=Control of Bus Accepted
				0-3		1-F=reserved
			04			Master Status Octet Mask
				7	1	Successful/Unsuccessful - Mandatory
				6	1	1=Bus Parity Error - Mandatory
				5		1=Pause
				4		1=Slave-Slave Operation Completed
				0-3	F	F=Defined Ending Status - Mandatory
			05			Slave Status Octet Mask
				7	1	Successful/Unsuccessful - Mandatory
				6	1	Bus Parity Error - Mandatory
				5		1=Pause
				4		1=Time Dependent Operation
				0-3	F	F=Defined Ending Status - Mandatory
			06			Request Modifier Octet Mask
				7		1=Facility Interrupts supported
				6	1	Report Busy Status - Mandatory
				5	1	Report Ready Status - Mandatory
				4	1	Power Fail Alert - Mandatory
				3	1	Power On Status Request - Mandatory
				0-2	7	Interrupt Class - Mandatory
			07			Selective Reset Control Octet Mask
				4-7	F	Slave Address - Mandatory
				3		1=Slave Release
				2		1=Re-initialize as at Power On
				1		1=Reset Logical Interface
				0		1=Reset Port Physical Interface
			08			Slave Interrupts Octet Mask
				7	0	reserved
				6	1	Report Busy Status - Mandatory
				5	1	Report Ready Status - Mandatory
				4		1=Priority Hold Status
				3	1	1=Priority Select Status
				0-2	7	Interrupt Class - Mandatory

(5) *Slave Status Octet Mask.* Bits 6-7 shall be set to 1, and the setting of the Bits 4-5 indicates that the slave supports these options at the Physical Interface. If Bits 0-3 are set to x'F', it indicates that the Encoded Ending status defined by Level 3 is supported. Support of defined Ending Status is required, and no other use of these bits may be made.

NOTE: If no Ending Status codes can be provided, "0000" shall be reported in the octet during the Ending Status sequence (see 4.10.4.1).

(6) *Request Modifier Octet Mask.* Bits 0-6 shall be set to 1. If Bit 7 is set to 1, it indicates that the Facility Interrupts Request option is supported.

(7) *Selective Reset Control Octet Mask.* Bits 4-7 shall be set to 1. If Bit 0 is set to 1, the slave shall set the condition of the Physical Interface port over which the Selective Reset was received to Neutral. If Bit 1 is set to 1, the slave shall reset its Logical Interface (e.g., microprocessor) upon a Selective Reset. If Bit 2 is set to 1, the slave shall be capable of reto the same conditions as at Power On (information in the slave of a transient or volatile nature would be cleared upon Selective Reset). If Bit 3 is set, the slave shall not perform a reset but shall release its drivers.

(8) *Slave Interrupts Octet Mask.* If this parameter is not present or contains all zeros, then Request Slave Interrupts is not supported by the slave.

6.3.4.13 Parameters 6C-6D. These parameters shall be as shown in Table 41.

6.3.4.13.1 Request Parm Parameter. This parameter may be used to control the transfer of parameters as data or to request that the slave respond with details on the parameters whose IDs are listed. If no IDs are listed, the slave shall respond with all of the parameters associated with the command.

(1) *Parameters as Data Bit.* When this bit is set, the slave shall return the requested parameters as data.

(2) *Parameters in Response Bit.* When this bit is set, the slave shall return the requested parameters appended to the response packet.

(3) *Length Bit.* When this bit is set, the slave shall return the accumulated length of the requested parameters in the Parm Length parameter of the response packet.

(4) *Naked Parameters as Data Bit.* When this bit is set, the slave shall return the requested parameter contents as data with no parameter information.

NOTE: Since there is no parsing information to identify parameters, it is advisable that the master request only one parameter to be returned in this manner, since there is no required order for the slave to return multiple parameters.

6.3.4.13.2 Parm Length Parameter. This parameter is returned in response to the Request Parm parameter that specified a number of IDs, and requested the slave to advise the accumulated length of same.

6.3.4.14 Parameter 6E (Slave Reconfiguration (Bit-Significant) Parameter). This parameter shall be as shown in Table 42. In its Initial attributes, the slave may identify a feature (e.g., Automatic Reallocation) that it supports. Such features may be turned "off" by the master.

The following bits are included in this parameter:

(1) *Report Conditional Success if Error Retry Bit.* When this bit is set to 1, the slave shall respond with Conditional Success Substatus if an error did not reoccur after retry. When this bit is set to 0, if the operation is completed without an error, either initially or upon retry, the slave shall report Successful.

(2) *Report Conditional Success if Data Correction Bit.* When this bit is set to 1, the slave shall respond with Conditional Success Substatus if a data error was corrected. When this bit is set to 0, if the operation is completed without an error, either initially or after correction, the slave shall report Successful.

(3) *Inhibit Extended Substatus in Response Bit.* When this bit is set to 1, the slave shall not transmit Extended substatus as part of the response packet.

(4) *Physical Selection of Synonyms Bit.* When this bit is set to 1, the slave shall recognize Synonyms in the range of 00-0F as Facility Addresses in the Select octet of the Physical Interface.

(5) *Automatic Reallocation On Bit.* This bit shall be set to 1 by the slave if it automatically reallocates blocks that require excessive retries, error correction, or both, or are otherwise identified by some optional analog algorithm. This capability should not be confused with defect mapping done during the FORMAT command. If the slave's Initial attributes show that it does not support Automatic Reallocation, and the master sets this bit, the slave shall reject this parameter setting.

Table 41
Attribute Parameters 6C-6D

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS	
M	n+1	6C				REQUEST PARM PARAMETER	
			01	7		Parameters as Data	*
				6		Parameters in Response	*
				5		Length	*
				4		Naked Parameters as Data	*
				3-0		reserved	
			02			Parameter ID	Repeated as many
			n			Parameter ID	times as needed
S	05	6D				PARM LENGTH PARAMETER	
			01-04			Length of Parameter List	

* Mutually exclusive parameters.

Table 42
Attribute Parameter 6E

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
B	n+1	6E				SLAVE RECONFIGURATION (BIT SIGNIFICANT)
			01	7		Report Conditional Success if Error Retry
				6		Report Conditional Success if Data Correction
				5		Inhibit Extended Substatus in Response
				4		Physical Selection of Synonyms
				3		Automatic Reallocation On
				2		Seek Algorithm On
				1		Inhibit Operation Response on Success
				0		reserved
			02	7		Transfer Notification Packets Required
				6		Inhibit Slave Messages
				5		Inhibit Unanticipated Pauses
				4		Disable All Error Recovery
				3		Log Unexpected Class 1 Events
				2		Discard Class 1 Condition Transitions
				1		Data Streaming Data Transfers
				0		Interlock Data Transfers
			03	7		Response on P-Busy to Not P-Busy Suppressed
				6-0		reserved

(6) *Seek Algorithm On Bit.* When this bit is set to 1, the slave shall reorder data access requests in order to minimize seek times (e.g., elevator algorithm). If the slave`s Initial attributes show that it does not support Seek Algorithms, and the master sets this bit, the slave shall reject this parameter setting.

(7) *Inhibit Operation Response on Success Bit.* When this bit is set to 1, the slave shall neither generate a Class 1 interrupt nor transmit a response packet to the master for a successfully completed command. This bit is generally used for data transfer commands to an unbuffered slave or facility. Response packets shall always be returned by the slave if the command did not complete sucessfully or if the command required slave/facility initiated recovery. If the slave does not allow the master to override the presentation of an Operation Response when a command completes successfully, it shall reject this parameter setting.

(8) *Transfer Notification Packets Required Bit.* When this bit is set to 1, the slave shall provide Transfer Notification packets, even if they are implicit (e.g., Facility Selection with Individual Commands). If the master sets this value to 0 in an attempt to override the presentation of Transfer Notification packets where they are required, the slave shall reject this parameter setting.

(9) *Inhibit Slave Messages Bit.* When this bit is set to 1, the slave shall not generate Message Exceptions.

(10) *Inhibit Unanticipated Pauses Bit.* When this bit is set, the slave shall not terminate a data transfer, except for anticipated pauses (burst boundaries), master pauses (if supported), or interface errors. If the slave can no longer transfer data, it shall remain in the XFRRDY state until it can transfer data or an error forces an unsuccessful completion of the data transfer.

(11) *Disable All Error Recovery Bit.* When this bit is set, the slave shall not invoke any error recovery in the execution of commands. All errors shall cause an immediate termination of the command with the error reported in Substatus.

(12) *Log Unexpected Class 1 Events Bit.* When this bit is set and there is no ANTICIPATED ACTION command queued, the slave shall record all unexpected Class 1 events (which are normally returned as Asynchronous or Message/Microcode Exception responses) in the Error Log, except Condition transitions (P-Available, Not Ready, and the like).

If the Error Log fills to the point that Error Log Request or Error Log Full substatus must be reported, all unexpected events shall be reported as Class 3 Interrupts to initiate prompt master action.

(13) *Discard Class 1 Condition Transitions Bit.* When this bit is set, the slave shall generate no interrupts or responses when such transitions occur. The master shall determine the condition of a slave or facility by the REPORT ADDRESSEE STATUS or ANTICIPATED ACTION commands.

(14) *Data Streaming Data Transfers Bit.* When this bit is set to 1, the slave shall transmit data in Data Streaming Mode. This setting is not needed unless the Physical Interface is set up for Interlocked mode.

(15) *Interlock Data Transfers.* When this bit is set to 1, the slave shall transmit interlocked data. This setting is not needed unless the Physical Interface is set up for Data Streaming mode.

(16) *Response on P-Busy to Not P-Busy Suppressed Bit.* When this bit is set, the slave shall generate Class 1 interrupts (but not the associated responses) when notifying the master that it is no longer busy after rejecting a selection sequence with a P-Busy response.

6.3.4.15 Parameter 6F (Slave Reconfiguration (Fields) Parameter). This parameter shall be as shown in Table 43. Fields that are not supported by the slave shall have all bits set to 1 (i.e., x'F...F'). If the master wishes to change only one field, all the others shall be set to x'F...F' and the slave shall ignore them (i.e., no existing value for a field shall be changed if the master set x'F...F'). The following octets are included in this parameter:

(1) *Setting of Time Dependency.* This value specifies the period that, if in the judgment of the slave would be exceeded prior to the start of data transfer, shall cause the slave to set the Time Dependent Operation bit at the Physical Interface on a Paused transfer.

(2) *Number of Queued Commands.* This value may be specified by the master to override the slave's setting of the maximum number of commands queued per facility. A master that cannot handle Transfer Notification packets can set this value to 1 to force commands to be Individual, and the slave shall not present Transfer Notification packets to the master. If this value is greater than 1, or the Transfer Notification Packets Required bit is set, the slave shall always present a Transfer Notification response prior to starting a data transfer.

(3) *Maximum Number of Octets in Command Packet.* This value specifies the maximum size for command packets.

(4) *Maximum Number of Octets in Response Packet.* This value specifies the maximum size for response packets.

(5) *Number of Unanswered SYNC INs during Data Streaming.* This value identifies the number of unanswered SYNC INs the slave transmits before it suspends transmission of SYNC IN's (i.e., if the number of SYNC INs minus the number of SYNC OUTs is equal to this attribute, the slave shall suspend SYNC IN transmission until SYNC INs minus SYNC OUTs is less than the maximum difference. (See Section 7 in ANSI X3.129-1986 for further details.)

(6) *Maximum Number of Multiplexed Data Transfers.* This value represents the maximum number of data transfers that the slave can multiplex at one time. A value of 0 is valid if the slave does not support multiplexed data transfers.

(7) *Generate Class 2 Interrupt.* The slave shall initiate a Class 2 Interrupt to indicate data transfer can begin when the value specified is ready to be transferred (In), or can be accepted (Out). The interrupt shall also be generated when the last portion of data to complete the command is ready to be transferred, even though the remainder may be less than the value specified.

Table 43
Attribute Parameter 6F

@	LTH	ID	OCTET	X/b	DEF	ATTRIBUTE PARAMETERS
B	n+1	6F				SLAVE RECONFIGURATION (FIELDS)
			01-04			Setting of Time Dependency (usecs)
			05-06			Number of Queued Commands
			07-08			Maximum Number of Octets in Command Packet
			09-0A			Maximum Number of Octets in Response Packet
			0B-0C			# of Unanswered SYNC INs during Data Streaming
			0D-0E			Max No of Multiplexed Data Transfers
			0F-12			Generate Class 2 Interrupt
			13-16			Burst Size
			17-1A			Data Streaming SYNC timeout (usecs)
			1B-1E			Port Release Time (usecs)
			1F-22			Facility Timeout (usecs)
			23-26			Implicit Release Delay (usecs)

PKT LTH	REF NO	OP CODE	COM MOD	OP MOD	SLAV ADDR	FAC ADDR	PARAMETERS
	0 1	2	3		4	5	6 through n
xxxx	xxxx	03	bbbb bbbb 7654 3210		xx	xx	

7654 3210
|||| Condition
||| Status
|| Port Query

Figure 26
Command Packet for REPORT ADDRESSEE STATUS

(8) *Burst Size.* This value (in octets) specifies the maximum size of data that a slave shall transfer during any one Information Transfer of data.

NOTE: The relationship between the buffer size when Class 2 is to be reported and the Burst Size defines the type of buffer management to be used by the slave.

(9) *Data Streaming SYNC Timeout.* This value specifies the period that the slave shall wait after SYNC OUTs cease from the master, before terminating the Information Transfer. If the value x'FFFFFFFE` is set, the time period shall be the maximum value allowed by the Physical Interface.

(10) *Port Release Time.* This value specifies the time period that a slave shall keep a port reserved without any activity following a PORT ADDRESS command, before implicitly releasing it. If a value of x'FFFFFFFE' is set, the time period shall be infinity.

(11) *Facility Timeout.* This value specifies the time period that a slave shall wait for a facility to perform an operation before terminating the command. If a value of x'FFFFFFFE` is set, the time period shall be infinity. One use of the timeout is to designate how long a facility may remain busy before the condition is reported to the master.

(12) *Implicit Release Delay.* This value specifies the time period that a slave shall wait for a master to respond with a selection sequence after a P-Busy to Not P-Busy interrupt is presented because the master had previously attempted selection. If a value of x'FFFFFFFE' is set, the time period shall be infinity.

NOTE: If two masters are competing for a port, one may be fast enough to release and then reselect before the other has time to respond, and this value can be set to give the slower master an opportunity to select the slave in such a situation.

6.4 REPORT ADDRESSEE STATUS

6.4.1 Command Packet. The command packet for this command shall be as shown in Figure 26.

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE	CODE 7	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 27
Response Packet for REPORT ADDRESSEE STATUS

Table 44
Report Addressee Status Parameters

@	LTH	ID	OCTET	X/b	DEF	REPORT ADDRESSEE STATUS PARAMETERS
B	02	50				PORT MASK PARAMETER
			01			Slave Address
			02			Facility Address
			03- n			Port Mask
S	n+1	51	01-02			CONDITION
			01	7		Operational
				6		Not Operational
				5		Ready
				4		Not Ready
				3		Facility Switched to Another Port
				2		Port Neutral
				1		L-Available
				0		P-Busy
			02	7		Status Pending
				6		Active
				5		Inactive
				4		P-Available
				3		Not P-Available
				2		reserved
				1		reserved
				0		reserved

6.4.2 Response Packet. The response packet for this command shall be as shown in Figure 27.

6.4.3 Description. REPORT ADDRESSEE STATUS shall cause the slave to report the condition, status, or port mask of the port or ports of the addressee. Execution of this command shall not clear any condition or status in the addressee.

One modifier bit is required for execution of this command, and the bits are mutually exclusive.

If the Condition modifier is set, the slave shall report the condition (see 4.7) of the addressee for the port or ports specified.

If the Status modifier is set, the slave shall report the status of the addressee. The response is a Vendor-Unique parameter that includes information about the addressee. One of the uses of this command is to permit a master that has not been available or operational to establish the configuration of operating slaves. Status is device type and implementation dependent. It includes the current status of the addressee (e.g., which side of an optical platter is loaded).

If the Port Query modifier is set, the slave shall report the addressee port mask(s) for the path over which the command was received. The master can use the Port Mask parameter(s) to find out which port, in a multi-ported slave configuration, it is connected to.

NOTE: The Path Control command provides the master with the capability of excluding other (presumed defective) masters from using a specific slave. The Port Mask defines the port that the master is connected to so that it can prevent excluding itself from port access.

6.4.4 REPORT ADDRESSEE STATUS Parameters. These parameters shall be as shown in Table 44.

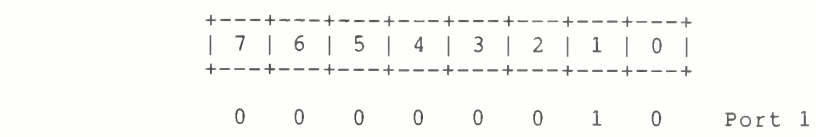


Figure 28
Port Mask Parameter

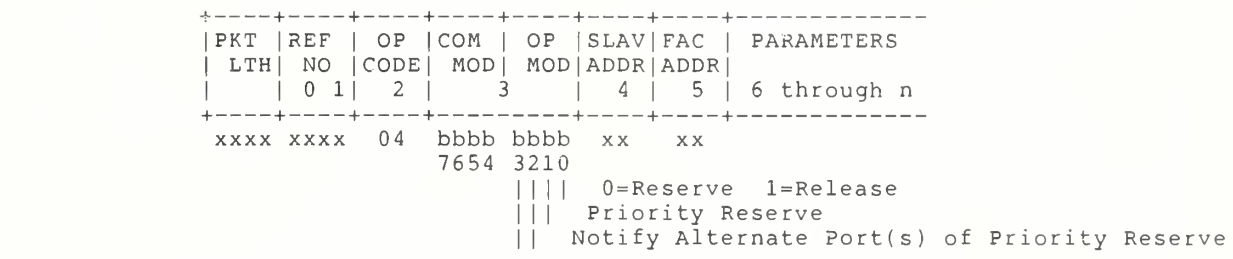


Figure 29
Command Packet for PORT ADDRESS

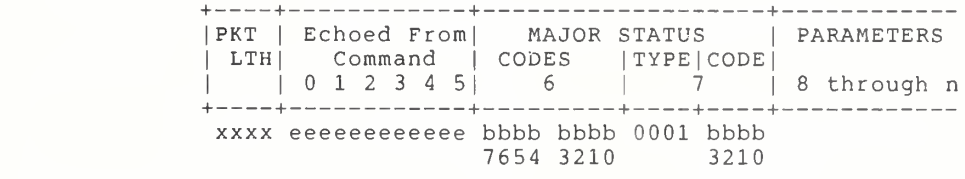


Figure 30
Response Packet for PORT ADDRESS

6.4.4.1 Port Mask Parameter. The first two fields in Figure 28 identify the slave or facility to which the parameter applies, and the bit(s) set in the mask identify the port(s).

When the Port Query modifier is set, this parameter identifies the port or ports on the path to the addressee over which the master issued the command.

If the master appends this parameter to the command, the slave shall repeat this parameter preceding each condition or status in the response for all the elements of the path.

A parameter length of 1 indicates that the command shall be performed for all the installed ports known to the addressee.

6.4.4.2 Condition Parameter. The two octets are bit significant to advise the master of the conditions within the addressee for the physical port or ports identified by the Port Mask parameter (i.e., the conditions at the Physical Interface as viewed by the master of that interface port).

6.5 PORT ADDRESS

6.5.1 Command Packet. The command packet for this command shall be as shown in Figure 29.

6.5.2 Response Packet. The response packet for this command shall be as shown in Figure 30.

6.5.3 Description. The PORT ADDRESS command allows the master to establish explicit allegiances with multiported slaves, facilities, or both.

If the Reserve modifier is set, it shall cause the addressee to be dedicated to the commanding port. The addressee shall remain reserved until a Release occurs or until another port issues a Priority Reserve or an appropriate reset is issued. Multiported addressees, when not reserved, shall be implemented to perform an implied reserve when executing a command from the master. Following the execution of the command or Chain, Sequence, or Order that caused the implied reserve, the addressee shall perform an implied release and become available to the other port or ports.

If the addressee supports command stacking, the addressee may be able to accept commands on one port while executing commands at another (this requires that each port of the addressee be capable of stacking commands). The addressee may defer the execution of stacked commands until the other port is no longer executing commands.

If more than one Reserve is issued to an addressee, succeeding Reserves shall be executed as NOPs. Slaves may employ a reserve safety timer to release the addressee if the reserving master does not access the addressee during the safety time limit that is specified by the Attributes. The safety time limit may be overridden by setting its value to infinity in the Attributes.

The Reserve of a facility shall cause it to be dedicated to the commanding port of the slave. However, the slave is not dedicated to the commanding master. When a command is issued to a multiported facility that is not reserved, the slave shall perform an implied reserve on the facility. At the completion of the operation, an implied release is issued to the facility. It is the slave's responsibility to bracket every operation to an unreserved, multiported facility with a reserve and a release function.

If the Relcase modifier is set, it shall cause the addressee to make itself available to its other port or ports. The Release complements a previous Reserve of the port. If multiple Reserves had been issued, it is not the responsibility of the slave to nest the Reserve/Release pairs to ensure symmetry. The first Release shall cause the addressee to make itself available to the other port or ports.

If the Priority Reserve modifier is set, the command shall not be accepted unless the Reserve modifier is also set (Bit 0=0). This modifier causes immediate switching of the addressee's port, releases any Access Permit ranges that may be active and establishes a Priority Reserve. If the addressee is engaged in a command through another port, it shall terminate the command, and return Priority reserved status to the other port. Termination may be either "graceful" or immediate, as determined by the slave/facility implementation. The Priority Reserve shall remain in effect until a Release command or an appropriate reset is issued.

Priority Reserve is primarily intended for failure recovery purposes. If the master attached to an unavailable port determines that the master on the active port has failed, the slaves, facilities, or both can be switched to the operable master through the use of the Priority Reserve command.

If a slave is capable of interpreting commands at an alternate port while reserved, Priority Reserve shall always be accepted, unless a port has been excluded by a previous PATH CONTROL command.

If the Notify Alternate Port(s) of Priority Reserve modifier is set, the slave shall advise the other port or ports by Asynchronous Response that this port has been Priority reserved. This modifier shall only be accepted by the slave if Bit 1 = 1 and Bit 0 = 0.

6.6 PATH CONTROL

6.6.1 Command Packet. The command packet for this command shall be as shown in Figure 31.

6.6.2 Response Packet. The response packet for this command shall be as shown in Figure 32.

6.6.3 Description. The PATH CONTROL command provides the master the ability to disable (Not P-Available), enable (P-Available), or assign any port of the addressee. The parameter shall be used as a mask to Disable (1) or Enable (0) specific ports on the addressee.

When there are no Operation Modifier bits set the ports shall be disabled in an orderly manner (i.e., after all activity for the port has ceased. The command is completed when all ports referenced in the mask are either enabled or successfully disabled.

There may be as many mask octets in the parameter as are needed to define the number of ports addressable (on a switching slave, this could be a large number, e.g., 32 ports would require 4 octets) (see Figure 33). The first port on a slave or facility is represented by bit 0 in the least significant octet and the remaining ports (if any) are represented by contiguous bits in the more significant octets.

If the Purge Commands Outstanding at Disabled Port modifier is set, it shall cause the slave to abort all commands currently in the command stack and clear all nonasynchronous responses for the disabled port of the addressee. The port or ports will be disabled regardless of the state of the disabled port. This command may be used to allow a malfunctioning port to be destructively disabled, regardless of its current condition.

Implementation Note: It is possible for an orderly disable to be executed against a slave port over which the PATH CONTROL command was received. If one is received, the command should be rejected.

```
+----+----+----+----+----+----+----+-------------
|PKT |REF | OP |COM | OP |SLAV|FAC | PARAMETERS
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR|
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+-------------
 xxxx xxxx  05  bbbb bbbb  xx   xx
                7654 3210
                     ||||  Purge Commands outstanding at Disabled port
                     |||  Path Select
```

Figure 31
Command Packet for PATH CONTROL

```
+----+------------+--------------------+------------
|PKT | Echoed From|    MAJOR STATUS    | PARAMETERS
| LTH|   Command  | CODES   |TYPE|CODE|
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+------------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 32
Response Packet for PATH CONTROL

```
+---+---+---+---+---+---+---+---+
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
+---+---+---+---+---+---+---+---+

  0   0   0   0   0   0   0   0    Enable all ports
  0   0   0   0   0   0   1   0    Disable Port 1/Enable Others
  0   0   0   0   1   0   0   1    Disable Ports 0 and 3/Enable Others
```

Figure 33
Mask Octet Examples

If the Path Select modifer is set and more than one Port Mask parameter is appended to the command, a group allegiance shall be established to the port or ports designated. The group allegiance remains until another PATH CONTROL command with this modifier set and only one Port Mask parameter appended is received, or an appropriate reset is executed. The command has the effect of reserving the addressee to the group of slave ports, facility ports, or both defined in the Port Mask parameters.

While the group allegiance is in effect, implicit and explicit allegiance from within a group work normally. A port excluded from the group allegiance may only gain control of the addressee through the use of Priority Reserve or an appropriate reset.

When the Path Select modifier is not set, but more than one slave Port Mask parameter is associated with a Port Mask parameter of the facility, the master is identifying alternate paths that may be used to route responses. Such a usage demands specific knowledge between the master and attached slaves, which may make the implementation vendor dependent, system dependent, or both.

Table 45
Path Control Parameter

@	LTH	ID	OCTET	X/b	DEF	PATH CONTROL PARAMETER
M	n+1	50				PORT MASK PARAMETER
			01			Slave Address
			02			Facility Address
			03- n			Octet mask(s)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6 through n	PARAMETERS
0006	xxxx	06	bbbb bbbb 7654 3210	xx	xx		

Bits 1, 2 have encoded meaning of:
x'0'=Enable x'2'=Disable
x'4'=Clear x'6'=Set

Figure 34
Command Packet for ATTENTION CONTROL

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 35
Response Packet for ATTENTION CONTROL

6.6.4 Path Control Parameter (Port Mask Parameter). This parameter is used to identify the port to be masked (see 6.4.1.1). This parameter shall be as shown in Table 45.

6.7 ATTENTION CONTROL

6.7.1 Command Packet. The command packet for this command shall be as shown in Figure 34.

6.7.2 Response Packet. The response packet for this command shall be as shown in Figure 35.

6.7.3 Description. The ATTENTION IN signal at the Physical Interface is set as a result of interrupts pending in the slave or slaves. The ATTENTION CONTROL command provides the master with the ability to control generation of the ATTENTION IN signal by the addressee. The modifiers are set in conjunction with the Attention Control Parameter to provide for the enabling, disabling, setting, or clearing the interrupt classes within the addressee.

This command does not affect the operation of other ports and does not affect the capability of the master to poll for interrupts at the Physical Interface. At Power On, all of the interrupt classes are enabled.

The Enable and Disable modifiers change how interrupts affect the setting of ATTENTION IN, as follows:

Enable allows the selected interrupts to generate attention (e.g., when Interrupt Mask Bit 2 is set, then Class 3 interrupts shall be enabled for causing the ATTENTION IN signal to be asserted).

Disable prevents the selected interrupts from generating attention (e.g., when Interrupt Mask Bit 0 is set, Class 1 interrupts shall not cause the ATTENTION IN signal to be asserted).

Table 46
Attention Control Parameter

@	LTH	ID	OCTET	X/b	DEF	ATTENTION CONTROL PARAMETER
M	02	50	01			INTERRUPTS MASK
				7-3		reserved
				2		Class 3 Interrrupts (Critical)
				1		Class 2 Interrrupts (Transfer Pending)
				0		Class 1 Interrrupts (Status Pending)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	07	bbbb bbbb 7654 3210	xx		xx	

0=Set 1=Report

Figure 36
Command Packet for OPERATING MODE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 37
Response Packet for OPERATING MODE

The Set and Clear modifiers force interrupts themselves to be changed, as follows:

Set causes the selected interrupts to be generated internally by the addressee (e.g., when Interrupt Mask Bit 0 is set, then a Class 1 interrupt shall be generated, which shall cause the assertion of the ATTENTION IN signal if Class 1 is Enabled).

Clear causes the selected interrupts to be reset (e.g., when Interrupt Mask Bit 0 is set, then the Class 1 interrupt indication shall be reset within the addressee).

6.7.4 Attention Control Parameter (Interrupts Mask Parameter). This parameter is used to set the mask to either cause or prevent interrupts from having the ATTENTION IN signal be generated.

This parameter shall be as shown in Table 46.

6.8 OPERATING MODE

6.8.1 Command Packet. The command packet for this command shall be as shown in Figure 36.

6.8.2 Response Packet. The response packet for this command shall be as shown in Figure 37.

6.8.3 Description. The OPERATING MODE command allows the master to change the operating modes of the slave or facility dynamically. The parameter field of the command packet defines what action the slave or facility is to take (e.g., those associated with establishing device-unique operating characteristics, such as recording density on tape, removal of media, and the like).

On disks, the master may direct transfer commands to nonprimary data spaces such as the CE Partition, IML Partition, and the like. Data transfer commands shall be Chained, Sequenced, or Ordered to OPERATING MODE with the Partition parameter in order to access the one desired.

Execution of transfer commands in slave-defined areas other than that for data may require different types of response information. The Response Conditions parameter is used to override the conditions established by Housekeeping Attributes.

The parameters associated with this command shall remain in effect until a subsequent Chained, Sequenced, or Ordered OPERATING MODE command is issued or until the termination of the Chain, Sequence, or Order.

6.8.4 Parameters. The Operating Mode Parameters shall be as shown in Table 47.

6.8.4.1 Partition (Common) Parameter. This parameter is used to establish the slave-defined Partition that is to be addressed when it is not the data partition.

6.8.4.2 Response Conditions Parameter. The master can override default attributes by use of this parameter, which has the same meanings as those defined in the Slave Reconfiguration Parameter of the ATTRIBUTES command (6.3.4.1.4).

6.8.4.3 Disk Modes Parameter

(1) *Spin Up.* The slave shall cause the disk spindle to be powered up and spinning, and shall unlock the carriage so that it will be ready for operation.

(2) *Spin Down.* The slave shall cause the disk spindle to be powered down.

(3) *Lock Cartridge.* The slave shall cause the disk cartridge or the cartridge door to be locked, so that a removable disk cartridge cannot be removed by an operator.

(4) *Unlock Cartridge.* The slave shall cause the disk cartridge or cartridge door to be unlocked, so that a removable cartridge can be removed by an operator.

(5) *Load Heads.* The slave shall cause the read/write heads to be loaded onto the media.

(6) *Unload Heads.* The slave shall cause the read/write heads to be unloaded from the media.

(7) *Lock Carriage.* The slave shall cause the disk carriage to be locked.

(8) *Rezero.* This parameter shall cause the arm of the addressee to be set to its initial calibrated position.

6.9 ABORT

6.9.1 Command Packet. The command packet for this command shall be as shown in Figure 38.

6.9.2 Response Packet. The response packet for this command shall be as shown in Figure 39.

6.9.3 Description. The ABORT command allows a master to terminate all or some of the commands within a slave or issue a Selective Reset to facilities.

Slaves shall be implemented to accept an ABORT command while commands are in progress on attached facilities. This requires that the slave have the ability to process a minimum of one ABORT command, even if its command stack is full. This is the only command that this restriction applies to, and only the simplest implementation need be supported when all queues are full.

Since an ABORT is presumed to take precedence over other commands, the Priority Modifier should be set; otherwise, it will be managed in the order of the slave's command handler (first in, first out (FIFO) or some other algorithm). The ABORT command shall be rejected if no modifiers are set and there are no parameters present.

ABORT allows the master to terminate commands in the slave or facility. Unless defined otherwise by the master via parameters, if any command in a Chain, Sequence, or Order is aborted, the remaining commands in the Chain, Sequence, or Order shall be terminated. It is the master's responsibility to ensure that no commands in a Chain, Sequence, or Order that had not been received by the addressee prior to issuing the ABORT are received following the ABORT.

The Command Reference Numbers of the commands to be terminated are passed as parameters. If no modifiers are set, the slave shall terminate all commands that have not been initiated. The response packets for the original commands shall identify whether or not the requested commands were terminated. The slave issues the Response for the ABORT command only after all of the commands identified in the parameters have been aborted.

If the Terminate Command in Progress modifier is set, the command in execution shall terminate immediately. If a Chain, Sequence, or Order of commands is affected, all other associated commands shall be terminated. Results are unpredictable if a command, command sequence, or command chain is partially completed at the time it is aborted. The slave shall abort the command in a manner that retains as much data or operational integrity as is practical.

Table 47
Operating Mode Parameters

@	LTH	ID	OCTET	X/b	DEF	OPERATING MODE PARAMETERS
B	n+1	3E				PARTITION PARAMETER (See 5.5.15)
M	02	50				RESPONSE CONDITIONS
			01	7		Post Conditional Success if Error Retry
				6		Post Conditional Success if Data Correction
				5		Inhibit Operation Response on Success
				4		Inhibit Extended Substatus Response
				3-0		reserved
M	03	51	01-02			disk MODES
			01	7		Spin Up
				6		Spin Down
				5		Lock Cartridge
				4		Unlock Cartridge
				3		Load Heads
				2		Unload Heads
				1		Lock Carriage
				0		Rezero
			02	7		
			to			reserved
				0		

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR	PARAMETERS 6 through n
xxxx	xxxx	08	bbbb bbbb 7654 3210	xx	xx		

```
                    ||||  Orderly Termination
                    |||  Terminate Command in Progress
                    ||  Terminate All Commands Not in Progress
```

Figure 38
Command Packet for ABORT

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 39
Response Packet for ABORT

If the Orderly Termination modifier is set, an in-progress command shall be terminated at a point that ensures data integrity. This may or may not require the slave to complete a sequence or chain.

If the Terminate All Commands Not in Progress modifier is set, all commands not in progress shall be aborted. A command may have been initiated but not be in progress (e.g., a transfer command with an implicit seek that has been completed, but no transfer has begun).

An ABORT command cannot be used to abort a previously issued ABORT command.

If a master is terminating commands it issued over the same port, short forms of the parameter list may be used. However, if on a multiported system, a different master (or the same master through a different port) issues the ABORT, the slave shall requires the complete routing information of the original commands. When the aborted command terminates, the port over which it was aborted is advised (i.e., the routing of the ABORT command overrides the original routing).

A master can issue a Selective Reset to a facility if the Facility Reset parameter is appended. The use of this parameter is mutually exclusive with the use of any operation modifiers or other parameters.

6.9.4 Parameters. The Abort Parameters shall be as shown in Table 48.

6.9.4.1 Command Reference Number Parameter. This parameter is used by the master that originally issued the commands to be aborted.

6.9.4.2 Alternate Port Commands Parameter. This parameter is issued by a master (which may or may not have issued the original command) for each command to be aborted, over a port other than that from which it was originally received. It is necessary to provide the slave with the complete routing information on the command.

(1) *Command Reference Number.* This value shall correspond to that of the command to be aborted.

(2) *Slave Address.* This octet shall identify the Slave Address to which the command was originally issued (slaves may have different addresses on different ports).

(3) *Facility Address.* This octet shall identify the Facility Address to which the command was originally issued (facilities may have different addresses on different ports).

(4) *Octet Mask(s).* This octet shall identify the port over which the command was originally received.

6.9.4.3 Facility Address Parameter. This parameter is used to remove commands queued to Facility Addresses. If there are several commands outstanding to a particular Facility Address, this parameter reduces the burden on the master to identify every Command Reference Number. The slave shall generate a response for each command aborted. This parameter is used as an alternative to Parameter 50.

6.9.4.4 Alternate Port Parameter. This parameter is issued by a master (which may or may not have issued the original command) for each command to be aborted, over a port other than that from which it was originally received. It provides the same function as the Facility Address parameter, with the addition of the routing information of Slave Address and port mask over which the commands were originally received.

If the Facility Address field is set to x'FF', then all of the commands received by that port shall be aborted. This parameter is used as an alternative to Parameter 51.

6.9.4.5 Facility Reset Parameter. When this parameter is appended all of the facility's pending and active commands and pending responses within the slave shall be flushed. Bits 3-0 (see 4.10.7) define the type of reset that the slave shall issue to the facility.

6.10 ACCESS PERMITS

6.10.1 Command Packet. The command packet for this command shall be as shown in Figure 40.

6.10.2 Response Packet. The response packet for this command shall be as shown in Figure 41.

6.10.3 Description. The ACCESS PERMITS command provides the ability to define multiple extents that may be either read- or write-protected by an Access Key established by the master. The Access Key is set originally by the master to be associated with a defined extent or extents, and specifies the type of protection desired.

Any command that references an extent or extents protected by a previously issued ACCESS PERMITS shall include the Access Key in its parameter list. If the Access Key is not present, or is incorrect, the slave shall terminate with Command Exception (Missing parameter) or Machine Exception (Access Violation) status, respectively.

Table 48
Abort Parameters

@	LTH	ID	OCTET	X/b	DEF	ABORT PARAMETERS	
M	n+1	50				COMMAND REFERENCE NUMBER PARAMETER	
			01-02			Command Reference Number	Repeated as many
			n-1:n			Command Reference Number	times as needed
M	n+1	51				ALTERNATE PORT COMMANDS PARAMETER	
			01-02			Command Reference Number	
			03			Slave Address	
			04			Facility Address	
			05- n			Octet mask(s)	
M	n+1	52				FACILITY ADDRESS PARAMETER	
			01			Facility Address	Repeated as many
			n			Facility Address	times as needed
M	n+1	53				ALTERNATE PORT PARAMETER	
			01			Slave Address	
			02			Facility Address	
			03- n			Octet masks	
M	02	54				FACILITY RESET PARAMETER	
			01	7-4		reserved	
				3		Release	
				2		Reset as Power On	
				1		Logical Interface Reset	
				0		Physical Interface Reset	

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	09	bbbb bbbb 7654 3210		xx	xx	

Bits 0, 1, 3 have encoded meaning of:
x'0'=Report
x'1'=Initialize x'2'=Restore
0=DataBlock 1=PhysicalBlock
x'9'=Load x'A'=Save

Figure 40
Command Packet for ACCESS PERMITS

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb	0001	bbbb7654 3210	3210

Figure 41
Response Packet for ACCESS PERMITS

The opcode modifiers allow the master to Initialize, Report, Restore, Load, or Save Access Key information. The modifiers are mutually exclusive, i.e., only one action may be specified by the command modifier.

NOTE: Bits 0, 1, and 3 (x'1', x'2', and x'8') are encoded. Bit 2 (x'4') is used in conjunction with these bits to refer to either logical or physical blocks.

The Opcode modifiers shall be as follows:

(1) *Report.* Report requires the addressee to respond with a list of the extents currently protected.

(2) *Initialize.* Initialize allows the slave to set its protected extents to their default value (typically none will be protected).

(3) *Restore.* Restore allows the restoration of Access Key information that has been Saved.

(4) *Save.* Save allows the addressee's protected extents, including those associated with this command, to be saved prior to Power Down or removal of the media from a removable media facility.

(5) *Load.* Load allows the master to set protected extents within the addressee (if they are valid).

(6) *Restore.* At Power On, slaves shall perform an automatic Restore if protected extents have been Saved; otherwise, they shall perform an Initialize.

The parameter fields associated with this command allow the protected data areas to be specified as extents, and use bit-significant fields to specify Read or Write Access, and to Invoke or Revoke protection. The maximum number of extents that can be protected is established with the ATTRIBUTES command using the Addressee Configuration parameter.

Protection may be revoked from selected areas by setting the Revoke bit in the parameter field. However, an ACCESS PERMITS command that references any part of a protected area shall be required to define the Access Key for that area. This is necessary to prevent accidental or deliberate revocation, modification, or remodification of access protected address ranges.

In multiport systems that have slaves that share common facilities, it is the responsibility of the master to ensure that appropriate protection is established for all slaves sharing them.

The protection is applied from the port that established the extents, and can be revoked by that port, or other ports identified in the original Protection parameter.

In the event that a master fails, or another master attached to a different port has a need to override existing ACCESS PERMITS, it is possible to redefine allowable access with the PORT ADDRESS command. All of the protected ACCESS PERMIT ranges associated with an addressee can be released simultaneously by issuing a PORT ADDRESS command with the Priority Reserve bit set in the Opcode Modifier field.

Implementation Note: The method specified for write-protected, or read-protected data, or both, is effective in preventing accidental access to address ranges. However, if all users of the system are allowed to issue a Priority Reserve (via a PORT ADDRESS command), then a malicious user can gain access to any data. This problem can be avoided if software restricts the use of the Priority Reserve bit in the PORT ADDRESS command.

6.10.4 Parameters. The Access Permits Parameters shall be as shown in Table 49.

6.10.4.1 Command Extent (Common) Parameter. This parameter is used to define the extent (or extents, if multiple parameters are supplied) that is to be protected by the slave, and is preceded by the Access Key and Access Protection parameters.

6.10.4.2 Response Extent (Common) Parameter. This parameter is used to return the extent or extents protected by the Access Key (which was supplied in the command packet).

6.10.4.3 Access Key (Common) Parameter. This parameter is used by the master to precede, and thus identify, the extent or extents to be protected or reported.

6.10.4.4 Data Address (Common) Parameter. This parameter is used if the four octets of Data Address in the Extent parameters are insufficient.

6.10.4.5 Partition (Common) Parameter. This parameter is used to address any partition other than the default data area, and shall precede the Extent parameter.

6.10.4.6 Port Mask Parameter (See 6.4.4.1). The Port Mask is used to establish from which ports the protection is to be applied. Applications of the Port Mask are used as follows:

(1) *This Port Only.* Used to protect the extent or extents on this port only.

(2) *This Master Only.* Used to protect the extent or extents accessible to this master only. It is necessary that the port controls be established to identify which master has access over which ports. A master may have more than one port of access to a slave.

Table 49
Access Permits Parameters

@	LTH	ID	OCTET	X/b	DEF	ACCESS PERMITS PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	50				PORT MASK PARAMETER
			01			Slave Address
			02			Facility Address
			03- n			Octet mask(s)
M	02	51				ACCESS PROTECTION PARAMETER
			01			Protection Modifiers
				7		Read Access
				6		Write Access
				5		Invoke * mutually
				4		Revoke * exclusive
				3-0		reserved

(3) *Other Ports Only.* Used to protect the extent or extents from access by the other ports connected to the slave.

(4) *Other Masters Only.* Used to protect the extent or extents from access by the other masters connected to the slave.

6.10.4.7 Access Protection Parameter (Protection Modifiers). These modifiers follow the Access Key parameter and are used to establish what kind of protection is needed for access to the extent. The modifiers are as follows:

(1) *Read Access.* This modifier permits transfer commands read-only access to the extent when associated with the correct key in their parameter list.

(2) *Write Access.* This modifier permits transfer commands both read and write access to the extent when associated with the correct key in their parameter list.

(3) *Invoke.* This modifier is used to establish, in accordance with the access modifiers, the protection of the extent or extents with the Access Key.

(4) *Revoke.* This modifier is used to cancel, in accordance with the access modifiers, the protection of the extent or extents under the Access Key.

6.11 RESUME

6.11.1 Command Packet. The command packet for this command shall be as shown in Figure 42.

6.11.2 Response Packet. The response packet for this command shall be as shown in Figure 43.

6.11.3 Description. This command allows the master, without reference to the termination address, to resume execution of an operation that was posted as Incomplete by the slave. The slave shall resume from the point at which the command had been suspended in execution, and continue as if there had been no interruption.

RESUME can only be used to reinitiate slave execution of the COPY, COMPARE SLAVE DATA, and COMPARE DATA commands. The response for the RESUME is presented when a command is successfully reinitiated. The reinitiated commands will each present a separate response when they complete.

For example, during a COPY operation from disk to tape, End of Media (EOM) may have occurred. Incomplete status would have been posted, and the master has to provide for rewind and replacement of the output tape. By issuing RESUME, the master can initiate resumption of the COPY without having to rethe COPY with revised disk extents information. The RESUME command thus

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	0A	bbbb bbbb 7654 3210		xx	xx	

Figure 42
Command Packet for RESUME

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 43
Response Packet for RESUME

Table 50
Resume Parameters

@	LTH	ID	OCTET	X/b	DEF	RESUME PARAMETERS
M	n+1	50	01-02			Command Reference Number
			n-1:n			Command Reference Number
M	n+1	51				ALTERNATE PORT COMMANDS PARAMETER
			01-02			Command Reference Number
			03			Slave Address
			04			Facility Address
			05- n			Octet masks

Repeated as many times as needed

facilitates the backup/restore operation when multiple backup media are required.

The command to be resumed shall be identified by its Command Reference Number, which is appended as a parameter (the Command Reference Number within the basic command packet is that for the RESUME command itself). If this parameter is missing or incorrect the slave shall terminate with Command Exception status.

If a slave is capable of executing simultaneously more than one command that can terminate with an Incomplete status, then the parameters listing the Command Reference Numbers of all those to be restarted shall be supplied as parameters. If a master is resuming commands it issued, only the short form of the parameter list is used.

If on a multiported system a different master (or the same master through a different port) issues the RESUME, the slave requires the complete routing information of the original command. When the resumed command terminates, the port over which it was resumed is advised, i.e., the routing of the resumed command overrides the original routing.

6.11.4 Parameters. The Resume Parameters shall be as shown in Table 50.

6.11.4.1 Command Reference Number Parameter. This parameter is used by the master that originally issued the command or commands to be resumed.

6.11.4.2 Alternate Port Commands Parameter. This parameter is issued by a master (which may or may not have issued the original command) for

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR	PARAMETERS 6 through n
xxxx	xxxx	0B	bbbb bbbb 7654 3210		xx	xx	

Figure 44
Command Packet for PORT RESPONSE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 45
Response Packet for PORT RESPONSE

Table 51
Port Response Parameters

@	LTH	ID	OCTET	X/b	DEF	PORT RESPONSE PARAMETERS
M	n+1	50				PORT MASK PARAMETER
			01			Slave Address
			02			Facility Address
			03- n			Octet mask(s)
M	n+1	51				RESPONSE INFORMATION PARAMETER
			01- n			Response Information

each command to be resumed, over a port other than that from which it was originally received. It is necessary to provide the slave with the complete routing information on the command.

(1) *Command Reference Number.* This value shall correspond to that of the command to be resumed.

(2) *Slave Address.* This octet shall identify the Slave Address to which the command was originally issued (slaves may have different addresses on different ports).

(3) *Facility Address.* This octet shall identify the Facility Address to which the command was originally issued (facilities may have different addresses on different ports).

(4) *Octet Mask.* This octet shall identify the port over which the command was originally received.

6.12 PORT RESPONSE

6.12.1 Command Packet. The command packet for this command shall be as shown in Figure 44.

6.12.2 Response Packet. The response packet for this command shall be as shown in Figure 45.

6.12.3 Description. The PORT RESPONSE command allows the master to request that the slave send an Asynchronous response to the port or ports specified by the Port Mask parameter. The response contains the Message/Microcode Substatus with the Port Response bit set.

6.12.4 Parameters. The Port Response Parameters shall be as shown in Table 51.

6.12.4.1 Port Mask Parameter. This parameter identifies the port or ports of the addressee that are to generate a response. If there is no parameter appended, the slave shall generate Asynchronous responses over all ports other than that over which the command was received (see also 6.4.4.1).

6.12.4.2 Response Information Transfer. This parameter contains the slave-independent information that will be contained in the Asynchronous response.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	PARAMETERS 7 through n
xxxx	xxxx	0C	bbbb bbbb 7654 3210		xx	xx	

Figure 46
Command Packet for ANTICIPATED ACTION

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 47
Response Packet for ANTICIPATED ACTION

Table 52
Anticipated Action Parameters

@	LTH	ID	OCTET	X/b	DEF	ANTICIPATED ACTION PARAMETERS
B	n+1	3D	01- n			ENCAPSULATION PARAMETER
M	03	50				EXPECTED CONDITIONS PARAMETER
			01	7		
				to		reserved
			02	2		
				1		Ready
				0		Not Ready

6.13 ANTICIPATED ACTION

6.13.1 Command Packet. The command packet for this command shall be as shown in Figure 46.

6.13.2 Response Packet. The response packet for this command shall be as shown in Figure 47.

6.13.3 Description. The ANTICIPATED ACTION command is normally issued when the master does not anticipate sending commands to the addressee for an extended period. It is used in conjunction with Attributes set regarding the slave action to be taken with unexpected Class 1 events (Asynchronous or Message/Microcode Exception responses).

This command provides the framework for a Command Completion response to return any unexpected conditions or status to the master. It shall be considered to complete successfully when an unexpected Class 1 event occurs and the Asynchronous or Message/Microcode Exception response shall be appended to the Command Completion Response in the Encapsulation Parameter.

6.13.4 Parameters. The Anticipated Action Parameters shall be as shown in Table 52.

6.13.4.1 Expected Conditions Parameter. This parameter identifies to the slave the conditions that the master expects to be in effect at the addressee when the command is executed. If the expected conditions are not met, the slave completes the command immediately with substatus relating to the addressee.

6.13.4.2 Encapsulation Parameter. This parameter is used to encapsulate the response packet that had been prepared because of an unexpected Class 1 event.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	41	bbbb bbbb 7654 3210	xx	xx		

Count 0=Octet 1=Block

0=DataBlock 1=PhysicalBlock

Figure 48
Command Packet for POSITION CONTROL

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 49
Response Packet for POSITION CONTROL

Table 53
Position Control Parameters

@	LTH	ID	OCTET	X/b	DEF	POSITION CONTROL PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

7. Position Commands

The commands in this section require the Command Extent parameter (Section 5) unless otherwise noted.

7.1 reserved (For use only in ANSI X3.147-1987)

7.2 POSITION CONTROL

7.2.1 Command Packet. The command packet for this command shall be as shown in Figure 48.

7.2.2 Response Packet. The response packet for this command shall be as shown in Figure 49.

7.2.3 Description. The POSITION CONTROL command causes the facility to be positioned according to the Extent parameter Data Address, which may be either logical or physical (e.g., on disk it may be CCHS (cylinder, head, sector)).

7.2.4 Parameters. The Position Control Parameters shall be as shown in Table 53.

7.2.4.1 Command Extent (Common) Parameter. This parameter is used to define the Data Address at which a disk is to be positioned. The value in the Count field shall be ignored for disks.

7.2.4.2 Response Extent (Common) Parameter. If the command fails, this parameter is used to return the Data Address of the last block accessed prior to failure.

7.2.4.3 Access Key (Common) Parameter. See 5.5.6.

7.2.4.4 Data Address (Common) Parameter. This parameter is used if the four octets of Data Address in the Extent parameters is insufficient.

7.2.4.5 Partition (Common) Parameter. This parameter is used to address other than the default data area.

7.3 REPORT POSITION

7.3.1 Command Packet. The command packet for this command shall be as shown in Figure 50.

7.3.2 Response Packet. The response packet for this command shall be as shown in Figure 51.

7.3.3 Description. The REPORT POSITION command instructs the addressee to report its current position. The position is returned in the Data Address field of the Response Extent parameter.

7.3.4 Parameters. The Report Position Parameters shall be as shown in Table 54.

7.3.4.1 Response Extent (Common) Parameter. This parameter is used to return the Data Address of the current position of the disk.

7.3.4.2 Access Key (Common) Parameter. See 5.5.6.

7.3.4.3 Data Address (Common) Parameter. See 5.5.11.

7.3.4.4 Partition (Common) Parameter. See 5.5.15.

7.4 reserved (For use only in ANSI X3.147-1987)

7.5 REPORT DISCONTINUITY

7.5.1 Command Packet. The command packet for this command shall be as shown in Figure 52.

7.5.2 Response Packet. The response packet for this command shall be as shown in Figure 53.

7.5.3 Description. The REPORT DISCONTINUITY command is used to provide the master with the location of time-dependent discontinuities (e.g., time-dependent track change, cylinder change, or reallocated defect) within the range specified in the Command Extent parameter.

The master may set more than one kind of discontinuity in the Stop on Discontinuity Type parameter. In response, the slave shall identify specifically the kind of discontinuity associated with the Data Address of the discontinuity.

If the Search modifier is set, the command shall terminate at the first incident of a specified discontinuity type.

If the List modifier is set, a list of all the locations within the extent that contain a time-dependent discontinuity shall be presented. If the list is expected to exceed the capacity of a response, then the master has to provide a Request Parms parameter to transfer as data.

The Stop on Discontinuity parameter shall precede the Data Address to which it refers.

Unless errors are encountered, this command shall terminate successfully either upon identifying discontinuities within the extent, or if there are none encountered.

7.5.4 Parameters. The Report Discontinuity Parameters shall be as shown in Table 55.

7.5.4.1 Command Extent (Common) Parameter. This parameter is used to define the extent that is to be checked by the slave for discontinuities.

7.5.4.2 Response Extent (Common) Parameter. This parameter is used to return the Data Address of the last block accessed prior to a positional discontinuity (e.g., reallocated block due to defect management) or a transition discontinuity (e.g., cylinder break between two blocks).

The Residual Count shall contain the value of the number of blocks remaining in the requested extent following the discontinuity.

7.5.4.3 Access Key (Common) Parameter. See 5.5.6.

7.5.4.4 Data Address (Common) Parameter. See 5.5.11.

7.5.4.5 Partition (Common) Parameter. See 5.5.15.

7.5.4.6 Stop on Discontinuity (Common) Parameter. This parameter is used to identify the type of discontinuity to be reported upon.

7.5.4.7 Request Parm Parameter. See 6.3.4.13.1.

7.5.4.8 Parm Length Parameter. See 6.3.4.13.2.

8. Transfer Commands

Transfer commands consist of data transfers, which may or may not include multiple blocks within or crossing physical boundaries. Upon recognizing a transfer command, the slave or facility positions to the Data Address specified in the Command Extent parameter, locates the block, and performs the required action. The command terminates when the amount of data specified by the Count has been transferred, or an error occurs that requires earlier termination. The status provided in the response packet shall identify what actions occurred during the execution of the command.

If a READ command is performed in the reverse direction, the first data octet read is the first data octet transfered to the master.

```
+----+----+----+----+----+----+----+-----------
|PKT |REF | OP |COM | OP |SLAV|FAC |  COMMAND
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR| PARAMETERS
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+-----------
 xxxx xxxx  42  bbbb bbbb  xx   xx
                7654 3210
                     ||||  Count 0=Octet 1=Block
                     |||
                     ||  0=DataBlock  1=PhysicalBlock
```

Figure 50
Command Packet for REPORT POSITION

```
+----+-----------+------------------+-----------
|PKT | Echoed From|   MAJOR STATUS   |  RESPONSE
| LTH|   Command  | CODES   |TYPE|CODE| PARAMETERS
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+-----------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 51
Response Packet for REPORT POSITION

```
+----+----+----+----+----+----+----+-----------
|PKT |REF | OP |COM | OP |SLAV|FAC |  COMMAND
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR| PARAMETERS
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+-----------
 xxxx xxxx  44  bbbb bbbb  xx   xx
                7654 3210
                     ||||  Count 0=Octet 1=Block
                     |||  0=Search  1=List
                     ||  0=DataBlock  1=PhysicalBlock
```

Figure 52
Command Packet for REPORT DISCONTINUITY

```
+----+-----------+------------------+-----------
|PKT | Echoed From|   MAJOR STATUS   |  RESPONSE
| LTH|   Command  | CODES   |TYPE|CODE| PARAMETERS
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+-----------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 53
Response Packet for REPORT DISCONTINUITY

Table 54
Report Position Parameters

@	LTH	ID	OCTET	X/b	DEF	REPORT POSITION PARAMETERS
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

Table 55
Report Discontinuity Parameters

@	LTH	ID	OCTET	X/b	DEF	REPORT DISCONTINUITY PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
B	07	3F	01- n			STOP ON DISCONTINUITY PARAMETER (See 5.5.16)
M	n+1	6C				REQUEST PARM PARAMETER (See 6.3.4.13.1)
			01	7		Parameters as Data *
				6		Parameters in Response *
				5		Length *
				4		Naked Parameters as Data *
				3-0		reserved
			02			Parameter ID Repeated as many
			n			Parameter ID times as needed
S	05	6D				PARM LENGTH PARAMETER (See 6.3.4.13.2)
			01-04			Length of Parameter List

* Mutually exclusive parameters.

8.1 READ

8.1.1 Command Packet. The command packet for this command shall be as shown in Figure 54.

8.1.2 Response Packet. The response packet for this command shall be as shown in Figure 55.

8.1.3 Description. The Direction modifier is not applicable to the disk.

The READ command transfers data from the addressee to the master starting at the location given in the Data Address of the Command Extent parameter. If positioning is required before the data can be accessed, the slave shall initiate the positioning operation. Refer to the POSITION CONTROL command for details of the positioning operation.

When the Data Recovery modifier is set to On, the slave or facility shall initiate data error recovery to attempt to recover data read with errors. If the error is unrecoverable, data shall be transferred up to the block in error and the command shall terminate. If the recovery attempts are successful, the transfer shall continue until all of the requested data has been transferred.

The response packet and associated parameters (if any) shall indicate the degree of recovery employed, depending on the default established by Attributes or a preceding OPERATING MODE command.

When the Data Recovery modifier is set to Off (data recovery disabled), the slave or facility shall terminate the command if a data error is detected. The data in error shall be transferred. Recovery from nondata transfer errors are not suppressed by the Data Recovery Off modifier.

When the access is complete the addressee shall verify correct access position and shall read data starting at the Data Address. The addressee shall transfer the number of sequentially addressed octets or blocks specified by the Count.

During a multiple block transfer on disks, if access boundaries are encountered, the facility shall perform appropriate access movements (unless overridden by a Boundary Gather parameter) and verify block positioning. The required processing, such as incrementing to the next PhysicalBlock head switches, seeking to the next cylinder when necessary, and restarting the read operation, shall be handled in a manner that does not require intervention or explicit instruction by the master.

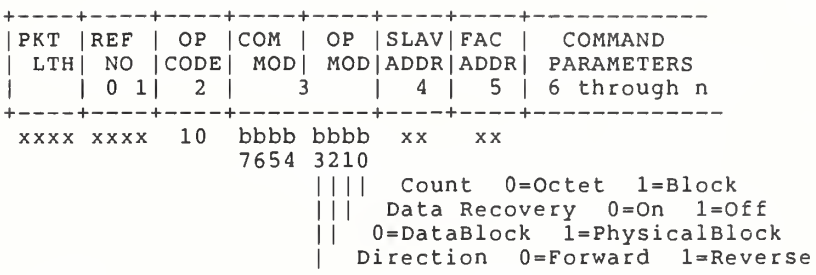


Figure 54
Command Packet for READ

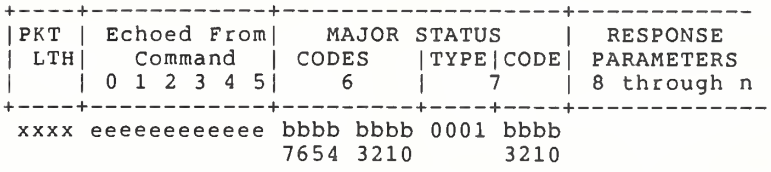


Figure 55
Response Packet for READ

If a defective block for which an alternate location has been assigned is encountered, the slave shall access the alternate block for reading, and then reestablish the access position for subsequent blocks to be read, if any. Access positioning shall be verified in all cases.

The slave shall present only valid data to the master if the Data Recovery modifier is set. All recovery from data errors detected during reading shall be attempted by the slave prior to transfer of the data to the master. If the data error is not corrected by the slave, processing of the command shall be terminated with a Machine Exception indicated in Major Status. The cause of the termination shall be indicated in substatus and extended substatus (if applicable).

The slave may permit execution of a "scatter read" if multiple Command Extent parameters are permitted (see Attribute Parameter 66 (6.3.4.8)). If multiple extents are accepted, the extents are processed by the slave in the sequence that they were received in the parameter list. Data is transferred to the master in the same manner as if the data was requested using a READ command with a single extent.

Should a command with multiple extents terminate prematurely due to data error or data recovery failure, the Response Extent shall contain the residual count of all nontransferred data within that extent. It is the master's responsibility to ascertain the total residual based on the Response Extent plus any other extents not processed.

A subset of "scatter read" defined as "boundary gather" allows the master to specify the starting address on a track, and have the slave perform an implicit scatter by not changing to the next track, but continuing transfer from the start of the track.

The use of multiple extents and the Skip Mask parameter are mutually exclusive.

8.1.4 Read Parameters

8.1.4.1 Common. These parameters shall be as shown in Table 56.

8.1.4.1.1 Command Extent (Common) Parameter. The Count specifies the number of blocks (or octets) to be transferred and cannot be zero. A zero value shall cause the command to be terminated with a Command Exception. The Data Address specifies the starting location. If the Data Address is not valid for the addressee, processing shall be terminated with Command Exception.

If the sum of the Data Address and the Count exceeds the number of blocks in the addressable area, processing of the command shall be terminated with Invalid Extent set in the Command Exception status.

8.1.4.1.2 Response Extent (Common) Parameter. This parameter is used to return the Residual Count of blocks (or octets) remaining in the transfer after it terminated. The Data Address varies depending on which modifiers had been set (see Implementation Note under 5.5.3).

8.1.4.1.3 Access Key (Common) Parameter. See 5.5.6.

8.1.4.1.4 Data Address (Common) Parameter. This parameter is used if the four octets of Data Address in the Extent parameters are insufficient.

8.1.4.1.5 Partition (Common) Parameter. This parameter is used to address other than the default data area.

8.1.4.1.6 Stop on Discontinuity (Common) Parameter. This parameter is used to advise the slave under which conditions to terminate a data transfer due to a timing delay due to a discontinuity.

8.1.4.2 Parameters 50-53. These parameters shall be as shown in Table 57.

8.1.4.2.1 Skip Mask Parameter. This parameter alters the action of this command from implicit transfers of consecutive blocks to the selective transfer of blocks under mask control. The parameter consists of a mask up to 32 octets in length. Each bit within the mask corresponds to a sequential DataBlock within the Command Extent.

NOTE: The number of 1 bits set in the mask shall equal the number of blocks specified in the Command Extent. Multiples of less than 8 blocks are padded up to a full octet.

DataBlocks are transferred if the corresponding mask bit value is "1", and are skipped if the value is "0". Bit 7 of the first octet of the mask corresponds to the Data Address in the Command Extent. Succeeding lower-order mask bits correspond to numerically higher valued addresses as illustrated below.

8.1.4.2.2 Information Transfer Size Override Parameter

(1) *Minimum Size of Data Transfers.* This value shall override the setting in Attributes for the duration of this command.

(2) *Maximum Size of Multiplexed Data Transfers.* This value shall override the setting in Attributes for the duration of this command.

8.1.4.2.3 Master Termination Permitted Parameter. This parameter allows the master to advise the slave that master termination may occur as a normal condition on this command, which is typically part of a chain. If the master does terminate transfer on a boundary, the slave shall not abort a Chain, Sequence, or Order. It shall respond with Master Terminated Transfer posted in Conditional Success status, and the Response Extent shall contain the residual count of the transfer.

If termination does not occur on a boundary, the slave shall abort the remaining commands of a Chain, Sequence, or Order and post the same status as if this parameter were not present, i.e., Data Length Difference posted in Incomplete Status.

8.1.4.2.4 Boundary Gather Parameter. This parameter shall immediately precede the Command Extent parameter.

During a multiple block transfer, if an access boundary is encountered , the facility shall not perform any access movement, but continue to transfer data until the Count specified has been transferred, or an error occurs that requires earlier termination.

8.2 READ RAW DATA

8.2.1 Command Packet. The command packet for this command shall be as shown in Figure 56.

8.2.2 Response Packet. The response packet for this command shall be as shown in Figure 57.

Table 56
Read Parameters - Common

@	LTH	ID	OCTET	X/b	DEF	READ PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	3F	01- n			STOP ON DISCONTINUITY PARAMETER (See 5.5.16)

Table 57
Read Parameters 50-53

@	LTH	ID	OCTET	X/b	DEF	READ PARAMETERS
M	n+1	50				SKIP MASK PARAMETER
			01	7		Reference to Block at Starting Address
				6		Reference to Block at Starting Address+1)
						. . . . .
				1		Reference to Block at (Ending Address-1)
			n	0		Reference to Block at Ending Address
M	n+1	51				INFORMATION TRANSFER SIZE OVERRIDE PARAMETER
			01-04			Generate Class 2 Interrupt
			05-08			Burst Size
M	01	52				MASTER TERMINATION PERMITTED PARAMETER
M	01	53				BOUNDARY GATHER PARAMETER

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx	xxxx	11	bbbb	bbbb	xx	xx	

```
7654 3210
|||| Count  0=Octet  1=Block
||| 1=Data Recovery Off
|| 0=DataBlock  1=PhysicalBlock
| Direction  0=Forward  1=Reverse
```

Figure 56
Command Packet for READ RAW DATA

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb	0001	bbbb	
		7654 3210		3210	

Figure 57
Response Packet for READ RAW DATA

Table 58
Read Raw Data Parameters

@	LTH	ID	OCTET	X/b	DEF	READ RAW DATA PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	3F	01- n			STOP ON DISCONTINUITY PARAMETER (See 5.5.16)

8.2.3 Description. The Direction modifier is not applicable to the disk.

The READ RAW DATA command reads data from the addressee, and transfers it to the master regardless of data errors encountered in the read. An error shall be reported only if the data cannot be transferred because of a condition such as inability to access the block, or because an unrecoverable slave or facility error (not associated with data transfer) is encountered.

The Data Recovery modifier shall be set to Off. The command shall be rejected if Bit 1 = 0.

8.2.4 Parameters. The Read Raw Data Parameters shall be as shown in Table 58.

8.2.4.1 Command Extent (Common) Parameter. See 5.5.2.

8.2.4.2 Response Extent (Common) Parameter. See 5.5.3.

8.2.4.3 Access Key (Common) Parameter. See 5.5.6.

8.2.4.4 Data Address (Common) Parameter. See 5.5.11.

8.2.4.5 Partition (Common) Parameter. See 5.5.15 and 8.1.4.1.5.

8.2.4.6 Stop on Discontinuity (Common) Parameter. See 5.5.16 and 8.1.4.1.6.

8.3 READ REPLICATED DATA

8.3.1 Command Packet. The command packet for this command shall be as shown in Figure 58.

8.3.2 Response Packet. The response packet for this command shall be as shown in Figure 59.

8.3.3 Description. The READ REPLICATED DATA command causes the addressee to read one instance of data that was written multiple contiguous times on a track. The intent of this command is to minimize latency in accessing data. The command directs the facility to transfer a string of *n* blocks from a range of *m* blocks, beginning at the Data Address in the Command Extent parameter or any block within range *m* whose address is the sum of the addressed block and an integral multiple of *n*.

Bit 0 shall be set to 1 for Block transfers only.

The Count field of the Command Extent parameter specifies the *n* value, and the Range Count parameter defines the number of blocks in the range *m*.

The facility shall initiate access to one of the repeated instances of *n* blocks (Extent Count) within the range *m* (Range Count), so as to minimize the access time to the data. When the access is complete, the facility shall verify the correct access position and transfer *n* blocks (Extent Count).

8.3.4 Parameters. The Read Replicated Data Parameters shall be as shown in Table 59.

8.3.4.1 Command Extent (Common) Parameter. See 5.5.2.

8.3.4.2 Response Extent (Common) Parameter. See 5.5.3.

8.3.4.3 Access Key (Common) Parameter. See 5.5.6.

8.3.4.4 Data Address (Common) Parameter. See 5.5.11.

8.3.4.5 Partition (Common) Parameter. See 5.5.15 and 8.1.4.1.5.

```
+----+----+----+----+----+----+----+------------
|PKT |REF | OP |COM | OP |SLAV|FAC |  COMMAND
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR| PARAMETERS
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+------------
 xxxx xxxx  12  bbbb bbbb  xx   xx
                7654 3210
                     ||||  1=Block
                     |||  Data Recovery  0=On  1=Off
                     ||  0=DataBlock  1=PhysicalBlock
```

Figure 58
Command Packet for READ REPLICATED DATA

```
+----+------------+-------------------+------------
|PKT | Echoed From|    MAJOR STATUS    |  RESPONSE
| LTH|   Command  | CODES   |TYPE|CODE| PARAMETERS
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+-------------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 59
Response Packet for READ REPLICATED DATA

Table 59
Read Replicated Data Parameters

@	LTH	ID	OCTET	X/b	DEF	READ REPLICATED DATA PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15 and 8.1.4.1.5)
M	05	50	01-04			RANGE COUNT PARAMETER

8.3.4.6 Range Count Parameter. This field specifies the number of blocks in the range. The range value shall be nonzero. If zero is specified, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. Invalid Parm Parameter shall identify the incorrect parameter.

If the Range Count is less than the Command Extent Count, or if the range is not an integral multiple of the Extent Count, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. An Invalid Parm parameter shall indentify the incorrect parameter.

If the sum of the Data Address and the Range Count exceeds the number of blocks in the partition, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. Invalid Parm Parameter shall identify the incorrect parameter.

8.4 SEARCH

8.4.1 Command Packet. The command packet for this command shall be as shown in Figure 60.

8.4.2 Response Packet. The response packet for this command shall be as shown in Figure 61.

8.4.3 Description. The Direction modifier is not applicable to the disk.

The SEARCH command causes the slave to search and logically compare the data strings supplied by the master (as parameters) with stored data. The search starts at the location specified by the Data Address, and continues until either a match is found to that supplied by the master (transfer begins with the block in which the match was found) or the Search Range is exhausted.

If data transfer occurred or the search was unsuccessful, the Response Extent is the same as on other transfer commands. However, if the string is located, but no data transfer was requested (selected by parameter setting), the Response Extent contains the address of the Data-Block containing the start of the string, and the Residual Count is unchanged.

The SEARCH command may be parameterized to perform three different types of search operations:

(1) A word-processing-style string search is obtained when one Set String Search Parameter is included in the command packet.

(2) A fixed location(s) data processing Key-type search is performed when the Set Multiple Key Search Parameter is included.

(3) A database-style search can be constructed by the use of multiple Set String Search parameters connected with the Boolean Operator parameter.

NOTE: The two parameters, Set String Search and Set Multiple Key Search are mutually exclusive and shall never be included in the same command packet.

The database-style search differs from a string search in that rather than searching for a single match condition, the parameters are used to construct a sequence of match conditions that must be met in order for the search to be satisfied. The slave searches for a match to the data model supplied by the master as Search String parameters joined by Boolean Operator parameters. The parameters have to be presented in the correct sequence. A multiple key search is counted or transferred only if the whole expression has been satisified successfully.

8.4.4 Search Parameters

8.4.4.1 Parameters 02, 31, 32, 3A, 3E. These parameters shall be as shown in Table 60.

8.4.4.1.1 Continuation of Preceding (Common) Parameter. See 5.2.1.7.

8.4.4.1.2 Command Extent (Common) Parameter. The Count specifies the number of blocks (or octets) to be transferred beginning at the first block that contained a match. This field may be set to zero if no data is to be transferred on a successful match. The Data Address specifies the starting location from which the command is to begin the search.

8.4.4.1.3 Response Extent (Common) Parameter. See 5.5.3.

8.4.4.1.4 Access Key (Common) Parameter. See 5.5.6.

8.4.4.1.5 Data Address (Common) Parameter. See 5.5.11.

8.4.4.1.6 Partition (Common) Parameter. See 5.5.15 and 8.1.4.1.5.

8.4.4.2 Parameter 50 (Set String Search Parameter). This parameter shall be as shown in Table 61 and shall be made up of the following octets:

(1) *Search Condition*. This octet specifies the conditions of the search.

When Bit 7 = 0, data shall be transferred to the master following a successful match condition (the number of octets/blocks to be transferred is specified by the Extent Count).

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	18	bbbb bbbb 7654 3210	xx		xx	

```
                    ||||  Count  0=Octet  1=Block
                    |||  Data Recovery  0=On  1=Off
                    ||  0=DataBlock  1=PhysicalBlock
                    |  Direction  0=Forward 1=Reverse
```

Figure 60
Command Packet for SEARCH

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 7	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 61
Response Packet for SEARCH

Table 60
Search Parameters 02, 31, 32, 3A, 3E

@	LTH	ID	OCTET	X/b	DEF	SEARCH PARAMETERS
M	01	02	01- n			CONTINUATION OF PRECEDING PARAMETER (See 5.2.1.7)
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

Table 61
Search Parameter 50

@	LTH	ID	OCTET	X/b	DEF	SEARCH PARAMETER
M	n+1	50				SET STRING SEARCH PARAMETER
			01	7		Transfer block(s)/Count number of matches
				6		reserved
				5		Search Greater Than
				4		Search Equal To
				3		Search Less Than
				2-0		reserved
			02			reserved
			03-06			Search Range
			07-0A			Offset into block to begin search
			0B- n			Data string to search against

Table 62
Search Parameter 51

@	LTH	ID	OCTET	X/b	DEF	SEARCH PARAMETER
M	03	51				BOOLEAN OPERATOR PARAMETER
			01	7		AND
				6		OR
				5		XOR
				4		NOT
				3		Begin Parentheses
				2		End Parentheses
				1-0		reserved
			02			Same definition as octet 01

When Bit 7 = 1, the slave shall count the number of successful match conditions.

If multiple Set Search String parameters are appended to a command, Bit 7 of each shall have the same setting as the first.

Bits 5-3 specify the kind of search operation to be performed. Comparisons between the supplied data string and stored data shall be performed by the slave on an octet-by-octet basis. The search operators are as follows:

Bits 5-3	Search Operators
000	invalid
001	search less than
010	search equal
011	search less than or equal
100	search greater than
101	search not equal
110	search greater than or equal
111	invalid

(2) *Search Range.* This field contains an unsigned binary number specifying a range of blocks, starting from the Data Address, over which the search operation is applied. The range value shall be nonzero. If zero is specified, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. An Invalid Parm parameter shall identify the incorrect parameter.

(3) *Offset into Block to Begin Search.* This field contains an unsigned binary number specifying an offset into the block to begin the search operation. This offset shall apply to all subsequent blocks in the Search Range. The value of this parameter shall be less than the block length. If the value is greater than, or equal to, the block length, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. Invalid Parm Parameter shall identify the incorrect parameter.

(4) *Data String.* This field of variable length (up to 240 octets in length) contains the search argument that shall be compared to stored data during the execution of the command.

8.4.4.3 Parameter 51 (Boolean Operator Parameter). This octet specifies the Boolean operators that can be used to join any two Set String Search parameters. Only one expression of AND, OR, or XOR can be supplied, and NOT can be associated with whichever is set to 1. The parentheses are used to allow correct Boolean expressions to be constructed, and one can be set in combination with any other bit.

If more than two bits are necessary to construct a Boolean expression, the second octet of the parameter is required (e.g., "AND NOT (" would require that bits 7 and 4 be set to 1 in octet 01 and bit 3 in octet 02).

This parameter shall be as shown in Table 62.

8.4.4.4 Parameter 52 (Set Multiple Key Search Parameter). This parameter shall be as shown in Table 63 and shall be made up of the following bits:

(1) *Search Condition* (See 8.4.4.2(1)). Bit 7 shall always be set equal to zero.

(2) *Search Range* (See 8.4.4.2(1)).

(3) *First Key Offset.* This value specifies an offset into the block to the location of the first Key field in which the search operation begins. This offset shall apply to all subsequent blocks in the Search Range. The value of this parameter shall be less than the block length. If the value is greater than, or equal to, the block length, processing of the request

Table 63
Search Parameter 52

@	LTH	ID	OCTET	X/b	DEF	SEARCH PARAMETER
M	n+1	52				SET MULTIPLE KEY SEARCH PARAMETER
			01	7	0	Transfer Blocks
				6		reserved
				5		Search Greater Than
				4		Search Equal To
				3		Search Less Than
				2-0		reserved
			02			reserved
			03-06			Search Range
			07-0A			First Key Offset
			0B-0E			Key Offset
			0F-12			Size of Keys - must be the same for all
			13-16			Number of Keys per block
			17- n			Search Key Argument

shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. Invalid Parm Parameter shall identify the incorrect parameter.

(4) *Key Offset.* This value specifies the offset between all Key fields in the block. The value of this parameter shall be nonzero and less than the block length. If the value is zero, or greater than or equal to the block length, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. An Invalid Parm parameter shall identify the incorrect parameter.

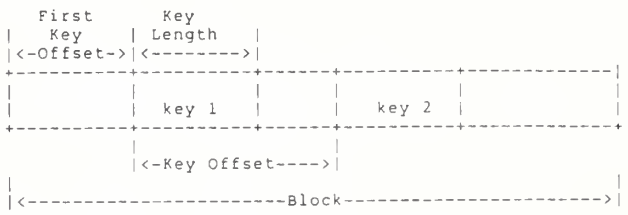
(5) *Size of Keys.* This value specifies the size of the Key field (shall be the same for all Keys). The value shall be nonzero and less than the block length. If the value is zero, or greater than or equal to the block length, processing of the request shall be terminated with a Command Exception indicated in Major Status, and Invalid Parameter shall be indicated in substatus. An Invalid Parm parameter shall identify the incorrect parameter.

(6) *Number of Keys per Block.* This value specifies the number of Key fields in a block, and shall be nonzero. If the value is zero, processing of the request shall be terminated with a Command Exception indicated in Major Status and, Invalid Parameter shall be indicated in substatus. An Invalid Parm Parameter shall identify the incorrect parameter.

(7) *Search Key Argument.* This variable length field contains the Search Key Argument that shall be compared with the Key fields within the blocks.

The search action specified in the Search Condition octet shall be repeatedly applied, block by block, between the Search Key Argument and data stored in Key fields within the blocks until either a match occurs or the Search Range has been exhausted.

Implementation Note: The following figure illustrates the relationship between all the fields of the Set Multiple Key Search Parameter.



8.4.4.5 Parameter 53 (Report Search Parameter). This parameter shall be as shown in Table 64 and consists of the following octets:

(1) *Search Results.* This octet contains the results of the search operation requested by this command. The search results shall be as follows:

Bits 5-3	Search Results
001	compared less than
010	compared equal
100	compared greater than

The results of a search not equal shall be indicated with either the less than or greater than bit setting. The remaining bits in this octet are reserved.

Table 64
Search Parameter 53

@	LTH	ID	OCTET	X/b	DEF	SEARCH PARAMETER
S	n+1	53				REPORT SEARCH PARAMETER
			01			Search Results
				7-6		reserved
				5		Compared Greater Than
				4		Compared Equal To
				3		Compared Less Than
				2-0		reserved
			02			reserved
			03-06			Offset of match into block
			07-0A			Count of blocks transferred/Number of matches
			0B-0E			Data Address of first match

(2) *Offset of Match into Block.* This field contains an unsigned binary number specifying the offset into the first block that contained the first octet of the match condition specified by the previous octet.

(3) *Count of Blocks Transferred/Number of Matches.* This field contains an unsigned binary number based upon the action requested in the Search Condition octet, i.e., the count of the number of blocks transferred to the master following a match, or a count of the number of matches detected within the Search Range (as per the setting of Bit 7 in the first octet of the parameter that set up the search).

(4) *Data Address of First Match.* This field is returned only on a Multiple Key Search and contains the Data Address of the first block that contained the match.

8.5 WRITE

8.5.1 Command Packet. The command packet for this command shall be as shown in Figure 62.

8.5.2 Response Packet. The response packet for this command shall be as shown in Figure 63.

8.5.3 Description. The Direction modifier is not applicable to the disk.

The WRITE command transfers data from the master to the addressee starting at the position specified in the Data Address of the Command Extent parameter. If positioning is required before the data can be accessed, the slave shall initiate the positioning operation. Refer to the POSITION CONTROL command (7.2) for details of the positioning operation. The mode and direction of the command is specified by the modifier octet.

If multiple Command extents are supported by the slave, they shall be processed in the sequence that they were received in the parameter list. Data is transferred to the slave since it was sequential data to be written to a single extent.

8.5.4 Parameters. The Write Parameters shall be as shown in Table 65.

8.5.4.1 Command Extent (Common) Parameter. See 5.5.2.

8.5.4.2 Response Extent (Common) Parameter. See 5.5.3.

8.5.4.3 Access Key (Common) Parameter. See 5.5.6.

8.5.4.4 Data Address (Common) Parameter. See 5.5.11.

8.5.4.5 Transfer (Common) Parameter. This parameter is used to specify the actions to be taken by the slave when writing to the media.

See Implementation Note at 5.5.13.

8.5.4.6 Partition (Common) Parameter. See 5.5.15.

8.5.4.7 Stop on Discontinuity (Common) Parameter. See 5.5.16.

8.5.4.8 Skip Mask Parameter. See 8.1.4.2.1.

8.5.4.9 Information Transfer Size Override Parameter. See 8.1.4.2.2.

8.5.4.10 Master Termination Permitted Parameter. See 8.1.4.2.3.

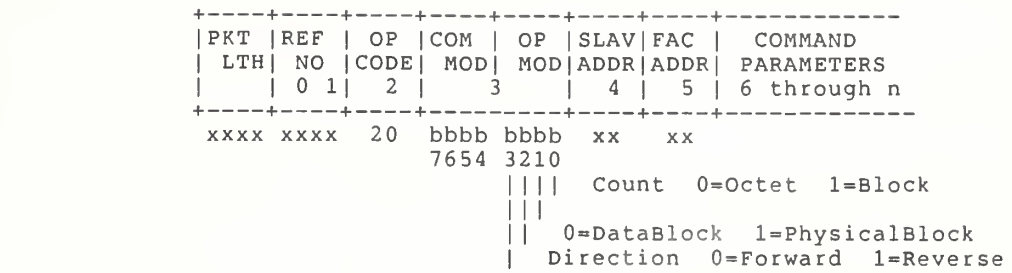


Figure 62
Command Packet for WRITE

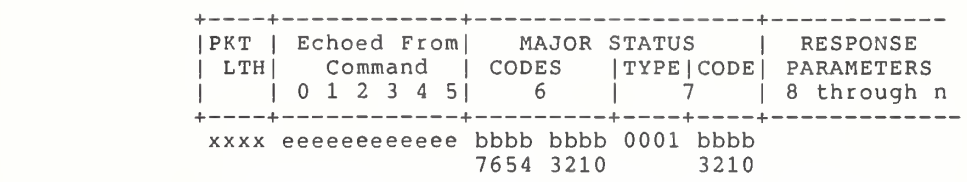


Figure 63
Response Packet for WRITE

Table 65
Write Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C				TRANSFER PARAMETER (See 5.5.13)
			01	7		Verify
				6		Volume
				5		Certify
				4		Stop on Data Error
				3-2		reserved
				1		Compare - Use buffer under slave control
				0		Compare - Master repeat transfer
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	3F	01- n			STOP ON DISCONTINUITY PARAMETER (See 5.5.16)
M	n+1	50	01- n			SKIP MASK PARAMETER (See 8.1.2.1)
M	n+1	51				INFORMATION TRANSFER SIZE OVERRIDE PARAMETER (See 8.1.2.2)
			01-04			Generate Class 2 Interrupt
			05-08			Burst Size
M	01	52				MASTER TERMINATION PERMITTED PARAMETER (See 8.1.2.3)

```
+----+----+----+----+----+----+----+------------
|PKT |REF | OP |COM | OP |SLAV|FAC |  COMMAND
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR| PARAMETERS
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+-------------
 xxxx xxxx  21  bbbb bbbb  xx   xx
                7654 3210
                     ||||  Count  0=Octet  1=Block
                     |||
                     ||  0=DataBlock  1=PhysicalBlock
                     |  Direction  0=Forward  1=Reverse
```

Figure 64
Command Packet for WRITE PATTERN

```
+----+------------+-------------------+------------
|PKT | Echoed From|    MAJOR STATUS    |  RESPONSE
| LTH|   Command  | CODES   |TYPE|CODE| PARAMETERS
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+-------------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 64A
Response Packet for WRITE PATTERN

8.6 WRITE PATTERN

8.6.1 Command Packet. The command packet for this command shall be as shown in Figure 64.

8.6.2 Response Packet. The response packet for this command shall be as shown in Figure 64A.

8.6.3 Description. The Direction modifier is not applicable to the disk.

The WRITE PATTERN command allows the master to write a data pattern in the extent specified by the Command Extent parameter. If a specific pattern is required by the master, it shall be passed as a parameter in the command packet. If the supplied data pattern is less than the number of blocks to be written, the supplied pattern shall be repeated as necessary to fill the blocks. If a Pattern parameter is not supplied, the command is either rejected or the Fill characters defined in Attribute 5A are used in lieu.

8.6.4 Parameters. The Write Pattern Parameters shall be as shown in Table 66.

8.6.4.1 Continuation of Preceding (Common) Parameter. See 5.2.1.7.

8.6.4.2 Command Extent (Common) Parameter. See 5.5.2.

8.6.4.3 Response Extent (Common) Parameter. See 5.5.3.

8.6.4.4 Access Key (Common) Parameter. See 5.5.6.

8.6.4.5 Data Address (Common) Parameter. See 5.5.11.

8.6.4.6 Transfer (Common) Parameter. See 5.5.13.

8.6.4.7 Partition (Common) Parameter. See 5.5.15.

8.6.4.8 Pattern Parameter. This parameter contains the pattern to be repeated. The length of the pattern cannot exceed the largest command packet size the slave can accept (as set in Attributes).

8.6.4.9 Request Parm Parameter. This parameter may be used to transfer a pattern as data.

Table 66
Write Pattern Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE PATTERN PARAMETERS
M	01	02	01- n			CONTINUATION OF PRECEDING PARAMETER (5.1.2.7)
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	50	01- n			PATTERN PARAMETER
M	n+1	6C	01- n			REQUEST PARM PARAMETER

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	28	bbbb 7654	bbbb 3210	xx	xx	

```
                    ||||  Count 0=Octet  1=Block
                    |||  Inhibit Defect Reallocation
                    ||  0=DataBlock  1=PhysicalBlock
                    |  1= Initialize Format
```

Figure 65
Command Packet for FORMAT

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE	CODE 7	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb 7654	bbbb 3210	0001	bbbb 3210

Figure 65A
Response Packet for FORMAT

8.7 FORMAT

8.7.1 Command Packet. The command packet for this command shall be as shown in Figure 65.

8.7.2 Response Packet. The response packet for this command shall be as shown in Figure 65A.

8.7.3 Description. The master can redefine the slave-defined Default Data Partition of a disk into smaller contiguous Partitions. There can only be one physical format to a Partition. The area to be formatted shall include the slave's allocation for alternate blocks.

The FORMAT command when used with the Initialize Format modifier set establishes a Partition and causes it be formatted. When the modifier is not set, the slave only formats the partition or extent defined by parameters. The formatting operation is slave specific, but the overall result is the establishment of Physical-Blocks through the defined Partition.

Typically, but not necessarily, formatting involves rewriting the identification and possibly the data fields of the disk. Some disk drives do

not have a simple relationship between PhysicalBlock, track, and cylinder addresses and it is the responsibility of the slave to format in accordance with the characteristics of the drives it is controlling.

If the Inhibit Defect Reallocation modifier is set, then the avoidance of defects by the slave is overridden. The default condition is for the slave to either skip or reallocate defects defined in the parameter list, in addition to those known to the slave by a stored defect list (if any). Defects encountered during the format shall also be skipped or reallocated. The techniques associated with how this is done are the responsibility of the vendor, and are not defined in this document.

If defect reallocation is not inhibited, entries in the Suspect Permanent and Suspect Temporary defect lists are automaticaly reallocated and moved to the corresponding Working defect list during execution of the FORMAT command.

It may be desirable for some applications to have the track format controlled by the master (e.g., a floppy disk used for media interchange between specific systems. Other applications are better handled by slave control of the formatting. In addition, some devices may not require defect mapping.

The Initialize Format modifier is used to indicate to the slave that this is an initial formatting of a Partition, rather than an incremental or update formatting. This may require special actions to be taken at the device level (e.g., disks implemented with a Level 2 IPI establish a Format Specification for the Partition, and need to know the difference between an initial and an incremental formatting.

The slave uses the Fill Octet field specified in Attributes as the data to be recorded within PhysicalBlocks during formatting.

Where a field is not supplied as a parameter value, the slave shall assume values that provide a viable format. The minimum information required by the slave is the DataBlock size, as the slave can select all other parameters necessary for formatting. The DataBlock size may be implicit if defined in Attributes prior to issuing the FORMAT command, or explicit by use of the DataBlock modifier and the Block Size Parameter. If the master is unable to support the maximum transfer rate of the slave the only other essential information would be the Transfer Rate Parameter.

Other use of master-supplied parameters presumes that the master has a knowledge of the disk, and the need to override the slave's selection of parameters in order to achieve a specific desired format.

When the slave is ready to begin the formatting operation, fields associated with the format that are in the addressee's Attribute Table shall have their contents set to zero by the addressee. Following a successful format, the addressee's Attribute Table shall be updated by the addressee to indicate the characteristics of the formatted media.

If a Maintenance Partition (ID x'01-0F') is to be formatted, it is necessary for the FORMAT command to be preceded in a Chain, Sequence, or Order by a OPERATING MODE command that specifies which Maintenance Partition is to be formatted. Some disk manufacturers physically compartment the disk into slave-defined Maintenance Partitions (ID x'01- 07') (e.g., a CE Cylinder that is provided to allow diagnostics to test read/write functions without affecting user data in the Data Partitions. Similarly, some masters may choose to define Maintenance Partitions (ID x'08-0F) for system diagnostic purposes.

If more than one partition is to be used on a disk, the slave requires as a minimum, the Count of the number of Blocks or octets in the Partition. If the DataBlock modifier is specified the slave shall begin the Partition in previously unformatted data space. If PhysicalBlocks are specified, the starting Physical Address is required, and shall begin on a physical boundary. The master shall precede all other supplied parameters with a Partition Parameter. For example,

Partition
    Blocksize
    Command Extent (Count only if DataBlocks)
            (Data Address required if
                PhysicalBlocks)
    Choice of Interleave Parameters

The burden for managing the implementation of partitions with different formats on the same disk is upon the master. Reference to partitions may be by either the Partition parameter or by Alias addressing.

8.7.4 Interleave Considerations. The order of blocks as seen by a master during data transfer is always in ascending block number sequence. Interleaving parameters are provided so that the

Table 67
Format Parameters 31, 32, 3A-3F

@	LTH	ID	OCTET	X/b	DEF	FORMAT PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	05	3B	01-04			BLOCK SIZE (See 5.5.12)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	3F	01- n			STOP ON DISCONTINUITY PARAMETER (See 5.5.16)

master can request that the order for recording of blocks on disk be in such a manner that blocks with sequentially ascending addresses are not physically contiguous. The effect of this is to reduce the effective transfer rate, though not the instantaneous transfer rate within a block. The degree by which blocks that are adjacent, but not physically contiguous, in the address space may have their transfer rate between the addressee and master effectively reduced is called the interleave factor.

The interleave factor allows an addressee with an effective data rate greater than that of the master, to be attached without incurring a significant performance loss. Noncontiguous physical fields are a commonly implemented technique, although other solutions that provide the same effect may be used also. An interleave factor may be established in manufacture, or it may be set by the master through parameters of the FORMAT command. Once formatted, the interleave factor becomes an attribute of the facility until changed by another FORMAT command. A reset or removal of power from the slave or facility shall not cause the interleave factor to change.

8.7.5 Defect List Considerations. If the master wishes to control the management of defects, it may provide defect list parameters for the slave to use in lieu of those known to the slave (by manufacturer's flaw map or slave's own maintained defect list).

If the list of defects supplied by the master exceeds the size of the command packet, then it may be supplied as data. The master is required to provide the information in the same manner as it wants the disk formatted (e.g., if the Block modifier is set, then it applies to both the extent to be formatted and the transfer of parameters as data.

The defect list supplied by the master is treated by the slave as applicable only during the execution of the command, and the slave makes no entries in its own defect lists. Typically, this method of formatting is used in lieu of slave management of defects. If the master wishes the defect list provided to be entered into the slave's defect list, it does so by using the Write Defect List command.

Typically, the master shall be formatting an area smaller than the whole disk, so the master supplies two Command Extent parameters, one for the transfer of parameters as data (see 5.2.4) and the other to define the area to be formatted.

8.7.6 Format Parameters

8.7.6.1 Parameters 31, 32, 3A-3F. These paramters shall be as shown in Table 67.

8.7.6.1.1 Command Extent (Common) Parameter. If this parameter is not present, the slave formats the Default Data Partition. If other partitions had been previously defined, the area formatted would be the remaining Default Data Partition; otherwise, the entire slave-defined Default Data Partition would be formatted.

If this parameter is present, the Count field establishes the size of the extent to be formatted. When the Initialize Format modifier is set, this also establishes the size of the partition.

If the DataBlock and Initialize Format modifiers are set, the Data Address field is not supplied, the partition is described in terms of DataBlocks, and the slave is responsible for allocating the partition into unformatted space of the Default Data Partition.

If the DataBlock modifier is set and the Initialize Format modifier is not, the Command Extent defines the area to be reformatted.

If the PhysicalBlock modifier is set, the Data Address field is required, the extent is described in terms of PhysicalBlocks, and the starting PhysicalBlock address shall be on a physical device boundary.

8.7.6.1.2 Response Extent (Common) Parameter. See 5.5.3.

8.7.6.1.3 Access Key (Common) Parameter. See 5.5.6.

8.7.6.1.4 Data Address (Common) Parameter. See 5.5.11.

8.7.6.1.5 Block Size (Common) Parameter. In the absence of this parameter, the slave shall select a size consistent with the DataBlock size defined in Attributes.

If this parameter is present and the Initialize Format modifier is set, it explicitly overrides any block size that may be defined in Attributes. The following actions are performed whether block size is implicit or explicit as long as the Initialize Format modifier is set.

If the DataBlock modifier is set, the block size is defined in DataBlocks for the partition, and the slave chooses a PhysicalBlock size that is consistent with the DataBlock size, and that optimizes the physical characteristics of the facility. The slave shall reject the FORMAT command with an Invalid Parm substatus if the DataBlock size is not supported for the Partition.

If the PhysicalBlock modifier is set, the block size is defined in PhysicalBlocks for the partition, and the slave checks this against any slave-defined or facility-defined constraints (e.g., switch settings, default values, or imbedded servo fixed sector disks. If the values do not match, the FORMAT command shall be rejected with an Invalid Parm Substatus.

8.7.6.1.6 Transfer (Common) Parameter. See 5.5.13.

8.7.6.1.7 Partition (Common) Parameter. See 5.5.15.

8.7.6.1.8 Stop On Discontinuity (Common) Parameter. Some methods of PhysicalBlock reallocation during formatting can cause discontinuities. The master uses this parameter if it wishes to have the FORMAT command cease executing because of an encountered discontinuity (see 5.5.16).

8.7.6.2 Parameters 50-55. These parameters shall be as shown in Table 68.

8.7.6.2.1 Number of PhysicalBlocks per Track Parameter. If this parameter is present, the value supplied specifies the number of PhysicalBlocks per track. If this parameter is not present, the slave determines the number.

8.7.6.2.2 PhysicalBlock Interleave Factors Parameter

(1) *Cylinder Interleave Factor*. This field specifies the number of PhysicalBlocks delay required between the last PhysicalBlock on the last track of a cylinder, and the first PhysicalBlock in the next cylinder. This delay biases the absolute location of PhysicalBlock 0 to compensate for the rotation of the disk during a single cylinder seek. A value of x'FF' allows the slave to determine the interleaving.

(2) *Head Interleave Factor*. This field specifies the number of PhysicalBlocks offset required between the last PhysicalBlock on a track, and the first PhysicalBlock in the next track. This delay biases the absolute location of PhysicalBlock 0 to compensate for the rotation of the disk during a head switch. A value of x'FF' allows the slave to determine the interleaving.

(3) *PhysicalBlock Interleave Factor*. This field is used to adjust the effective transfer rate to and from the disk. It specifies the number of PhysicalBlocks to be skipped before the next PhysicalBlock is generated. A value of zero causes no skip, and the PhysicalBlocks are sequentially numbered. A value of x'FF' allows the slave to determine the interleaving. The slave shall generate interleaved PhysicalBlocks, while also accomodating track and cylinder interleaving.

Table 68
Format Parameters 50-55

@	LTH	ID	OCTET	X/b	DEF	FORMAT PARAMETERS	
M	03	50	01-02			NUMBER OF PhysicalBlocks PER TRACK	
M	04	51				PhysicalBlock INTERLEAVE FACTORS	
			01		FF	Cylinder Interleave Factor	
			02		FF	Head Interleave Factor	
			03		FF	PhysicalBlock Interleave Factor	*
M	n+1	52	01-n			PhysicalBlock INTERLEAVE TABLE	*
M	n+1	53				TRANSFER RATE (Octets/second)	*
			01-04			Effective disk Transfer Rate	
			05-08			Master's Instantaneous Transfer Rate	
M	03	54				DataBlock INTERLEAVE PARAMETER	
			01	7	1	Value	
				6-0	0	reserved	
			02			DataBlock Interleave Value	*
M	03	54				DataBlock INTERLEAVE PARAMETER	
			01	7	0	Factor	
				6-0	0	reserved	
			02			DataBlock Interleave Factor	*

* Mutually exclusive parameters.

8.7.6.2.3 PhysicalBlock Interleave Table Parameter. This parameter allows the master to describe the PhysicalBlock interleave structure for a complete track. The number of entries in the table shall be equal to the number of PhysicalBlocks in a track. The first entry in the table specifies the physical PhysicalBlock number to be given to absolute PhysicalBlock 0 on each track; the second entry gives the physical PhysicalBlock number to be assigned to absolute sector 1; and so on. This parameter can be used to achieve a level of software interleaving (e.g., if the operating system software is implemented in block sizes of 1K, but the media is formatted with PhysicalBlock of 512 octets, the table for a simple example of 8 PhysicalBlocks per track could be constructed as '12563478'.

8.7.6.2.4 Transfer Rate Parameter. The first field is a value that allows the slave to format the disk in whichever fashion it chooses to achieve an effective data rate equal to or less than that specified. The second field is used by the slave to adjust its internal data flow to match the instantaneous transfer rate to and from the master.

8.7.6.2.5 DataBlock Interleave Parameter. This paramter is made up of either a Value or a Factor octet.

(1) *Value*. When Bit 7 of the first octet is set to 1, the Value defines the interleave value to be used in formatting the media.

The Value specifies one of the interleave values defined by the facility. Bit 0 specifies interleave value 0, which is the basic block transfer capability of the facility. Bits 1-7 specify interleave values, which are defined by vendor specifications. Individual bits 1-7 are set to 1 to cause a reduction in the block transfer rate that exceeds the reduction of the immediately preceding bit (e.g., Bit 5 = 1 shall cause a greater reduction in the block transfer rate then Bit 4 = 1). Only one bit may be set.

(2) *Factor*. When Bit 7 of the first octet is set to 0, the Factor specifies the factor to be used in formatting the media.

The Factor is used to adjust the effective rate at which DataBlocks are transferred from the disk. It specifies the number of DataBlocks to be skipped before the next sequential DataBlock is generated. A value of zero causes no skip, and a value of x'FF' allows the slave to determine the interleaving.

Table 69
Format Parameters 56-59

@	LTH	ID	OCTET	X/b	DEF	FORMAT PARAMETERS	
M	n+1	56				TRACK DEFECTS LIST	
			01-04			Cylinder	(first)
			05-06			Track	
			07-0A			Octet Offset into track	
			0B-0C			Length of defect (in bits)	
			n-B:8			Cylinder	(last)
			n-7:6			Track	
			n-5:2			Octet Offset into track	
			n-1:n			Length of defect (in bits)	
M	n+1	57				SECTOR DEFECTS LIST	
			01-04			Cylinder	(first)
			05-06			Track	
			07-08			No of Sector after Index	
			09-0A			Octet Offset within Sector	
			0B-0C			Length of defect (in bits)	
			n-B:8			Cylinder	(last)
			n-7:6			Track	
			n-5:4			No of Sector after Index	
			n-3:2			Octet Offset within Sector	
			n-1:n			Length of defect (in bits)	
M	06	58				HARD DISK FORMATS	
			01-02			Number of Alternate Cylinders	
			03			Number of On-track PhysicalBlock Spares	
M	06	59				FLOPPY FORMATS	
			01			Standard Formats	
				7-4		reserved	
				3		ANSI 8" SS SD 128 octets/PhysicalBlock	
				2		ANSI 8" SS DD 256 octets/PhysicalBlock	
				1		ANSI 8" DS DS 256 octets/PhysicalBlock	
				0		ANSI 8" DS Quad Density 1024 octets/PhysBlock	
			02			Number of Heads	
			03-04			Gap Length	
			05			Fill Octet	

8.7.6.3 Parameters 56-59. These parameters shall be as shown in Table 69.

8.7.6.3.1 Track Defects List Parameter. This parameter allows specifying the location and size of a list of defects by cylinder number, head number, and offset from index-by-octet location. This parameter is used for disks without fixed sectors.

8.7.6.3.2 Sector Defects List Parameter. This parameter is used to identify defects in fixed sector disks (e.g., imbedded servo) by their position within the sector, as located by cylinder number, head number, and sector number.

8.7.6.3.3 Hard Disk Formats Parameter. This parameter contains the following bits:

(1) *Number of Alternate Cylinders.* This field specifies the number of cylinders to be taken for use in allocating alternate tracks, alternate PhysicalBlocks, or both.

(2) *Number of On-Track PhysicalBlock Spares.* This field specifies the number of PhysicalBlocks to be reserved on each track for assigning alternates for those PhysicalBlocks that contain defects. If this field's value is zero, there are no spares, and alternates shall be taken from the Alternate Cylinders. On-track PhysicalBlock spares typically decrease the capacity of the disk but increase performance because the access time penalty associated with defective PhysicalBlocks mapped to Alternate Cylinders is avoided.

8.7.6.3.4 Floppy Formats Parameter. This paramter is made up of the following bits:

(1) *Standard Formats.* This field is used to identify to which standard format the floppy is to be formatted. If more than one bit is set, the parameter is rejected; and if no bits are set, the format is nonstandard and the remaining parameter fields are required.

(2) *Number of Heads.* This value specifies the number of heads (i.e., sides) on the floppy to be formatted.

(3) *Gap Length.* This value specifies the number of octets between PhysicalBlocks when a floppy disk is to be formatted. It is only used if a nonstandard floppy format is to be generated.

Table 70
Format Parameters 5A, 6C

@	LTH	ID	OCTET	X/b	DEF	FORMAT PARAMETERS	
M	n+1	5A				CELL DEFECTS LIST PARAMETER	
			01-04			Cylinder	(first)
			05-06			Track	
			07-08			No of Defective Cells	
			09-0C			Offset of Defective Cell #1	
			0D- m			Offset of Defective Cell #2 through end	
			n -			Cylinder	(last)
			n -			Track	
			n -			No of Defective Cells	
			n -			Offset of Defective Cell #1	
			n -			Offset of Defective Cell #2 through end	
M	n+1	6C				REQUEST PARM PARAMETER	
			01	7		Parameters as Data	*
				6		Parameters in Response	*
				5		Length	*
				4		Naked Parameters as Data	*
				3-0		reserved	
			02			Parameter ID	Repeated as many
			n			Parameter ID	times as needed

* Mutually exclusive parameters.

8.7.6.4 Parameters 5A, 6C. These parameters shall be as shown in Table 70.

8.7.6.4.1 Cell Defects List Parameter. This parameter allows specifying the location and size of a list of defects by cylinder number, head number, and offset of cell from index. This parameter is used for disks with variable record sizes (e.g., IPI-2 Format 2). The No of Defective Cells is used to step through the list within each cylinder (4 * No of Defective Cells identifies location of next cylinder's information).

8.7.6.4.2 Request Parm Parameter. See 6.3.4.13.1.

9. Combination Commands

The commands in this section are the most complex commands of the Logical Interface. These commands involve considerations that by their very nature may be device-specific,vendor-specific, or both. Refer to vendor specifications as to their implementation.

Some Combination commands do not require Combination Extents. Command Completions that do not contain a Combination Extent are identified as Transfer Responses (0001), not Combination Responses.

The Combination commands are typically addressed to the slave. However, the command packet may be addressed to a facility; in which case, the Facility Address in each Extent parameter shall be the same as that of the addressee.

On Combination commands that execute as part of a Chain, the Source and Destination addressees shall be the same as the remainder of the chain.

NOTE: Combination commands do not require the use of the Physical Interface Slave-to-Slave Information Transfers if the facilities are attached to the same slave. Users should be aware that if Physical Interface Slave-to-Slave Information Transfers are to be used, then dedication of the interfaces is required.

It is not intended that Slave-to-Slave Information Transfers execute concurrently with other operations. With Slave-to-Slave Information Transfers, the dominant slave shall have all of the capabilities of a master, such as the handling of unexpected responses, even though it does not have control over selection.

9.1 COPY

9.1.1 Command Packet. The command packet for this command shall be as shown in Figure 66.

9.1.2 Response Packet. The response packet for this command shall be as shown in Figure 67.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	30	bbbb bbbb		xx	xx	

Figure 66
Command Packet for COPY

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 67
Response Packet for COPY

9.1.3 Description. The COPY command allows the transfer of data between slaves (if Slave-to-Slave Information Transfers is supported at the Physical Interface Level 1), or between facilities attached to the slave.

If supported by the slave, multiple extents may be used for both the source and destination. The command requires that at least one source and one destination extent be defined in Combination Extent parameters; additional extents may be supplied and are processed by the slave in the sequence that they were received. The number of combination commands that may be active in a slave at the same time is defined by attributes.

If the two facilities have different block sizes, the transfer length is expressed in terms of the block size used by the source. The block size of the source shall be converted by the slave to the block size of the destination, as required. On the write to the last destination block, the destination addressee shall perform an "update write" operation; i.e., if the transfer of the last source block does not fill the destination block, the slave or facility shall not alter the remainder of the destination block.

It is possible for either the source or the destination, and sometimes both, to contain a Residual Count in the response packet (e.g., if the command terminated due to error, or end of tape on a multireel file was encountered (see also the RESUME command)).

Each set of COPY parameters is assumed to refer to an extent on the source and destination facilities. A continuous extent on a disk is equivalent to a span of blocks. COPY parameters may be concatenated so that multiple extents may be transferred via a single command packet. This allows files consisting of multiple noncontiguous extents to be backed up or restored with little overhead in the master.

If, for example, a file is to be restored from a backup tape (which can be treated as one large logically contiguous segment), then the master shall know the available storage areas on the destination drive and their lengths. One set of COPY parameters is generated for each logically contiguous area on the destination disk.

An entire disk/tape/floppy may be transferred with a single COPY command by setting the Volume modifier bit in the Transfer parameter. However, unless the two facilities involved in the COPY operation have the same capacity, the transfer shall be terminated whenever the lower-capacity facility reaches its limit.

The process terminates when one of the following conditions is encountered:

(1) The specified extents have been transferred for all sets of COPY parameters in the packet (nominal termination)

(2) A file mark is read from a source tape

(3) The end of tape is found

Table 71
Copy Parameters

@	LTH	ID	OCTET	X/b	DEF	COPY PARAMETERS
M	n+1	33				COMBINATION COMMAND EXTENT PARM (See 5.5.4)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Count
			09-0C			Data Address
			n-7:4			Count — Repeated as many
			n-3:n			Data Address — times as needed
S	n+1	34				COMBINATION RESPONSE EXTENT PARM (See 5.5.5)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Residual Count
			09-0C			Data Address
			0D-0E			Major Status
			0F- n			Substatus if any - Codes x0-xB
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	02	50	01			EXTENDED MODIFIERS
				7		Suppress Automatic Retries
				6		Transfer If Error *
				5		Skip Bad PhysicalBlock
				4		Terminate on Unrecoverable Error *
				3-0		Reserved

* Mutually exclusive modifiers.

(4) The source disk is empty; i.e., last block of the source disk has been read and transferred to the destination

(5) The destination disk or tape is full; i.e., last block of the destination disk has been written

The COPY response packet shall define the completion status. If the COPY operation has terminated prematurely (i.e., before all specified data has been transferred), but successfully, it is considered Incomplete and may be resumed from the point of termination with the RESUME command.

In the case of premature termination, the master shall be sent the appropriate response packet defining which facility caused the termination. It is then the master's responsibility to initiate a RESUME command to restart from the point of termination. This mechanism allows the master to prompt the operator for another tape cassette or floppy disk before the operation is continued.

9.1.4 Parameters. The Copy Parameters shall be as shown in Table 71.

9.1.4.1 Combination Command Extent (Common) Parameter. This parameter is used to identify the extent as being a Source or Destination. The Slave and Facility Addresses are required because the dominant slave responsible for execution of the command may or may not be either the source or the destination.

The modifiers in Bits 0-3 are those that would normally be command modifiers for a READ or WRITE.

9.1.4.2 Combination Response Extent (Common) Parameter. This parameter is used to report the Residual Count on the Slave and Facility Source and Destination extents. The Substatus associated with the Source or Destination addressee that was in use at the time the command terminated is also presented.

9.1.4.3 Access Key (Common) Parameter. This parameter is used to provide the key that allows access to a protected area.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	31	bbbb bbbb		xx	xx	

Figure 68
Command Packet for COMPARE SLAVE DATA

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 7	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 69
Response Packet for COMPARE SLAVE DATA

9.1.4.4 Data Address (Common) Parameter. This parameter is used if the four octets of Data Address in the Extent parameters is insufficient.

9.1.4.5 Transfer (Common) Parameter. This parameter is used to specify the actions to be taken by the slave when writing to the media.

9.1.4.6 Partition (Common) Parameter. This parameter is used to address other than the default data area.

9.1.4.7 Extended Modifiers Parameter. This parameter is needed only if the master has a need to override the normal error handling and correction features of the slave, and are associated only with the source. Such a situation can occur when the media source has been physically damaged, or otherwise affected in some way that affects its ability to be read. The following bits may be set when an error is encountered:

(1) *Suppress Automatic Retries.* Automatic retries are suppressed when an error is encountered; i.e., the error action shall be taken on the first error.

CAUTION: If retries are suppressed, then any error will leave a gap in the data on the destination device. Setting this bit makes the COPY operation vulnerable to soft errors. If not set, automatic retries will be attempted before any error action is taken.

(2) *Transfer if Error.* PhysicalBlocks containing ECC errors shall be transferred with the error intact (e.g., the ECC code has not been updated so the error would occur when the destination block is accessed again). It is the responsibility of the master to ensure that the ECC codes of the source and the destination are the same.

If this bit is not set, then the ECC shall be updated on the destination block and the error will not be detectable on the destination unit.

(3) *Skip Bad PhysicalBlocks.* PhysicalBlocks that cannot be read correctly shall be ignored; i.e., treat that PhysicalBlock as if it had never been read. The Residual Count is not decremented and the data is not transferred. If not set, then the error action is specified by Terminate on Error.

(4) *Terminate on Unrecoverable Error.* The COPY operation shall be terminated if an unrecoverable error occurs. The appropriate error status shall be reported to the master.

9.2 COMPARE SLAVE DATA

9.2.1 Command Packet. The command packet for this command shall be as shown in Figure 68.

9.2.2 Response Packet. The response packet for this command shall be as shown in Figure 69.

Table 72
Compare Slave Data Parameters

@	LTH	ID	OCTET	X/b	DEF	COMPARE SLAVE DATA PARAMETERS
M	n+1	33				COMBINATION COMMAND EXTENT PARM (See 5.5.4)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Count
			09-0C			Data Address
			n-7:4			Count — Repeated as many
			n-3:n			Data Address — times as needed
S	n+1	34				COMBINATION RESPONSE EXTENT PARM (See 5.5.5)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Residual Count
			09-0C			Data Address
			0D-0E			Major Status
			0F- n			Substatus if any - Codes x0-xB
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	02	50	01			EXTENDED MODIFIERS
				7		Suppress Automatic Retries
				6-5		reserved
				4		Terminate on Unrecoverable Error
				3-0		reserved

9.2.3 Description. The COMPARE SLAVE DATA command compares data on one facility with data on another facility attached to the same slave, or compares two sets of data on the same facility. The data on the source is the reference for the comparison and its Count determines the length of the comparison.

The command terminates when all data has been compared or a data inequality is encountered. If the two facilities have different block sizes, the Count shall be expressed as the number of blocks required for the data on the source. The slave shall perform the mapping between the two block sizes.

If the destination data space is smaller than the source. Incomplete status is posted. The Residual Count of the source may be used to calculate the amount of data actually compared and the Data Addresses shall contain the address of the last block compared.

The source and destination may be of multiple extents if parameters are appended to the command packet.

9.2.4 Description. The Compare Slave Data Parameters shall be as shown in Table 72.

9.2.4.1 Combination Command Extent (Common) Parameter. See 5.5.4 and also 9.1.4.1.

9.2.4.2 Combination Response Extent (Common) Parameter. See 5.5.5 and also 9.1.4.2.

9.2.4.3 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.2.4.4 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.2.4.5 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.6.

9.2.4.6 Extended Modifiers Parameter. See 9.1.4.7.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	32	bbbb	bbbb	xx	xx	

Figure 70
Command Packet for COMPARE DATA

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 7	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 71
Response Packet for COMPARE DATA

9.3 COMPARE DATA

9.3.1 Command Packet. The command packet for this command shall be as shown in Figure 70.

9.3.2 Response Packet. The response packet for this command shall be as shown in Figure 71.

9.3.3 Description. The COMPARE DATA command compares data on the facility with data from the master. The Count at the source (master) is defined by the Command Extent parameter, and determines the length of the comparison. The Count shall be expressed as the number of blocks required for the data on the source.

The destination shall be defined by a Combination Extent parameter. The slave shall perform any mapping necessary between the data received and the block size at the destination.

The command terminates when all the data to be compared has been received from the master, a data inequality is encountered, or end of extent (Incomplete status) occurs on the destination. The Residual Count in the source's Response Extent parameter can be used to calculate the amount of data actually compared. The Data Address in the destination's Combination Extent response shall contain the address of the last block in which a comparison was made.

9.3.4 Parameters. The Compare Data Parameters shall be as shown in Table 73.

9.3.4.1 Command Extent (Common)Parameter. See 5.5.2.

9.3.4.2 Response Extent (Common) Parameter. See 5.5.3.

9.3.4.3 Combination Command Extent (Common) Parameter. See 5.5.4 and also 9.1.4.1.

9.3.4.4 Combination Response Extent (Common) Parameter. See 5.5.5 and also 9.1.4.2.

9.3.4.5 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.3.4.6 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.3.4.7 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.6.

9.3.4.8 Extended Modifiers Parameter. See 9.1.7.

9.4 REALLOCATE

9.4.1 Command Packet. The command packet for this command shall be as shown in Figure 72.

9.4.2 Response Packet. The response packet for this command shall be as shown in Figure 73.

9.4.3 Description. The REALLOCATE command causes defective recording areas to be reassigned to replacement recording areas. The address of the recording areas to be reallocated is specified by the Address field within an Extent parameter, in a Defect List parameter, or in a Suspect defect list.

Table 73
Compare Data Parameters

@	LTH	ID	OCTET	X/b	DEF	COMPARE DATA PARAMETERS
M	05	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
S	05	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
M	n+1	33				COMBINATION COMMAND EXTENT PARM (See 5.5.4)
			01			Slave Address
			02			Facility Address
			03			Modifiers (Bit 7 = 1)
			04			Reserved
			05-08			Count
			09-0C			Data Address
			n-7:4			Count — Repeated as many
			n-3:n			Data Address — times as needed
S	n+1	34				COMBINATION RESPONSE EXTENT PARM (See 5.5.5)
			01			Slave Address
			02			Facility Address
			03			Modifiers (Bit 7 = 1)
			04			Reserved
			05-08			Residual Count
			09-0C			Data Address
			0D-0E			Major Status
			0F- n			Substatus if any - Codes x0-xB
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	02	50	01			EXTENDED MODIFIERS
				7		Suppress Automatic Retries
				6-5		reserved
				4		Terminate on Unrecoverable Error
				3-0		reserved

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	33	bbbb (7654)	bbbb (3210)	xx	xx	

|||| Relocate Data

Figure 72
Command Packet for REALLOCATE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE	CODE 7	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb (7654) bbbb (3210)	0011	bbbb (3210)	

Figure 73
Response Packet for REALLOCATE

The Data Address is implicitly interpreted by the slave, unless the Data Address parameter is used to specify which kind of blocks (logical, physical, or absolute) are to be defined.

If the Relocate Data modifier is set, the slave shall copy data from the specified recording area on the facility to a replacement recording area on the same facility. The destination may be explicit as specified by a destination Extent, or implicit if the slave has the capability to automatically manage the alternate area.

The original recording area shall be marked as unusable and the replacement recording area shall be automatically referenced by the facility whenever the original Data Address is accessed.

Slaves may only support this command in a predefined mode of operation, in which case only the Command Extent and Response parameters are required to identify the block being reallocated.

The execution of REALLOCATE with no defects specified as appended parameters causes defects listed in either the Suspect Permanent or Suspect Temporary defect lists to be reallocated.

If any defects are specified as appended parameters, then they are the only ones that will be reallocated, and no reference shall be made to the Suspect lists.

The locations of defective blocks are returned to the command that encountered the defect, and these are the addresses used for the REALLOCATE command. This command may be used in either DataBlock or PhysicalBlock mode. If DataBlock reallocation is selected, the slave shall determine the PhysicalBlocks affected by the defective recording area. The defective recording area and as many additional recording areas as are necessary to accommodate the DataBlock shall be reallocated.

If the space required to reallocate the specified block is not available (e.g., all space assigned for that purpose has been exhausted), the command shall terminate with Reallocation Space Exhausted status.

NOTE: Some facility technologies may not provide for a master-requested reallocation of a data area. Normal completion status in the response packet indicates that the slave executed the command according to its internal algorithm, as specified in the functional specification of the product.

9.4.4 Reallocate Parameters

9.4.4.1 Parameters 31-3E. These parameters shall be as shown in Table 74.

9.4.4.1.1 Command Extent (Common) Parameter. See 5.5.2.

9.4.4.1.2 Response Extent (Common) Parameter. See 5.5.3.

9.4.4.1.3 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.4.4.1.4 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.4.4.1.5 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.6.

9.4.4.2 Parameters 50, 55. These parameters shall be as shown in Table 75.

9.4.4.2.1 Defect Parameter. This parameter may be used instead of a source extent, and the defect shall be added to the Defect List.

(1) *Octet Offset.* The Octet Offset is relative to the location identified in the Data Address parameter (required to precede this parameter).

(2) *Length of Defect.* The Length of Defect is the size of the defect counted in bits.

9.4.4.2.2 Defective DataBlock Parameter. This parameter is used by the master to supply a list of defects identified by DataBlock address to the slave. A master that uses only DataBlock addresses has no other reference and presumably these were identified by slave responses following data transfers.

The list shall be processed in such a way that the following sequence of operations can be used to reinstate the pattern of media substitutions that was in place when a previous Temporary defect list was read:

(1) Execute an initializing FORMAT without changing the DataBlock size to remove Temporary defects

(2) Execute a REALLOCATE command with a Defective DataBlock parameter that contains a previously read Temporary defect list in chronological order

This should be true as long as the DataBlock size was not changed, and no Permanent defects were added since the defect list was read.

The defective DataBlocks shall be categorized and identified by using the following octets:

(1) *Size of Datablock Address Fields.* This field specifies the size of the addresses. The field shall be a multiple of 16 bits, and be a minimum of 4 octets.

(2) *Address of Defective DataBlock.* A list of fields containing the addresses of DataBlocks that contain defects.

Table 74
Reallocate Parameters 31-3E

@	LTH	ID	OCTET	X/b	DEF	REALLOCATE PARAMETERS
M	n+1	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

Table 75
Reallocate Parameters 50, 55

@	LTH	ID	OCTET	X/b	DEF	REALLOCATE PARAMETERS	
M	n+1	50				DEFECT PARAMETER	
			01-04			Octet Offset	
			05-06			Length of Defect (in bits)	
			n-5:2			Octet Offset	Repeated as many
			n-1:n			Length of Defect	times as needed
M	n+1	55				DEFECTIVE DATABLOCK PARAMETER	
			01-02			Size of Datablock Address Fields	
			03-			Address of Defective DataBlock	repeated as
			n- :n			Address of Defective DataBlock	many times needed

9.5 ALLOCATE RESTORE

9.5.1 Command Packet. The command packet for this command shall be as shown in Figure 74.

9.5.2 Response Packet. The response packet for this command shall be as shown in Figure 75.

9.5.3 Description. The ALLOCATE RESTORE command allows a previously reallocated recording area to be restored and to be marked as usable.

If the Relocate Data modifier is set, the data in the allocated replacement recording area shall be restored to the original recording area. The address of the restored recording area shall be removed from the facility's defect list. The address of the previously reallocated recording area is transferred as either data or parameters.

On disks, if a DataBlock is contained within a PhysicalBlock, the contents of the replacement PhysicalBlock will be copied to the original PhysicalBlock and the defect linkage shall be cleared. If the DataBlock requires several PhysicalBlocks, the contents of replacement PhysicalBlocks used by the DataBlock will be copied to the original PhysicalBlocks, and the defect linkage shall be cleared.

This command reverses the effect of the REALLOCATE command. It would typically be used if an error condition within the addressee caused an abnormal number of data reallocations (e.g., marginal read/write board).

The master is required to effectively mirror the original REALLOCATE commands and parameters so that the slave can execute this command successfully. Considerations of interleaving and multiple reallocations may require that ALLOCATE RESTORE commands be received in the reverse order that the REALLOCATE commands were issued.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	PARAMETERS 6 through n
xxxx	xxxx	34	bbbb bbbb 7654 3210 ||||	Relocate Data	xx	xx	

Figure 74
Command Packet for ALLOCATE RESTORE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 75
Response Packet for ALLOCATE RESTORE

9.5.4 Allocate Restore Parameters

9.5.4.1 Parameters 31-35, 3A, 3E. These parameters shall be as shown in Table 76.

9.5.4.1.1 Command Extent (Common) Parameter. See 5.5.2.

9.5.4.1.2 Response Extent (Common) Parameter. See 5.5.3.

9.5.4.1.3 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.5.4.1.4 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.5.4.1.5 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.6.

9.5.4.2 Parameters 50, 55. These parameters shall be as shown in Table 77.

9.5.4.2.1 Defect Parameter. This parameter may be used instead of a source extent to remove a defect from the Defect List. See 9.4.4.2.1 for definition of fields.

9.5.4.2.2 Defective DataBlock Parameter. See 9.4.4.2.2.

9.6 SHADOW READ

9.6.1 Command Packet. The command packet for this command shall be as shown in Figure 76.

9.6.2 Respose Packet. The response packet for this command shall be as shown in Figure 77.

9.6.3 Description. The SHADOW READ command transfers data from the primary or secondary addressee starting at the location given in the Data Addresses. Whenever a SHADOW READ operation is invoked, the slave shall determine from which addressee the data will be transferred. In a multiported environment, the slave shall determine which addressee is available and initiate the data transfer operation. A slave may also optimize the I/O operation based on optimum head and cylinder geometry.

The command executes in one of two modes depending on whether or not shadowing is on a file basis or by volume.

When executing on a file basis, the master shall specify the Count and Data Address for each Extent parameter (the PhysicalBlock sizes may be different for each facility).

When executing on a volume basis, only one Count and Data Address need be specified, since the extent on the second facility is a mirror of the first facility.

Table 76
Allocate Restore Parameters 31-35, 3A, 3E

@	LTH	ID	OCTET	X/b	DEF	ALLOCATE RESTORE PARAMETERS
M	n+1	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

Table 77
Allocate Restore Parameters 50, 55

@	LTH	ID	OCTET	X/b	DEF	ALLOCATE RESTORE PARAMETERS	
M	n+1	50				DEFECT PARAMETER	
			01-04			Octet Offset	
			05-06			Length of Defect (in bits)	
			n-5:2			Octet Offset	Repeated as many
			n-1:n			Length of Defect	times as needed
M	n+1	55				DEFECTIVE DATABLOCK PARAMETER	
			01-02			Size of Datablock Address Fields	
			03-			Address of Defective DataBlock	repeated as
			n- :n			Address of Defective DataBlock	many times
							as needed

PKT LTH	REF NO	OP CODE	COM MOD	OP MOD	SLAV ADDR	FAC ADDR	PARAMETERS
	0 1	2	3		4	5	6 through n
xxxx	xxxx	35	bbbb bbbb		xx	xx	

Figure 76
Command Packet for SHADOW READ

PKT LTH	Echoed From Command	MAJOR STATUS CODES	TYPE	CODE	PARAMETERS
	0 1 2 3 4 5	6	7		8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 77
Response Packet for SHADOW READ

Table 78
Shadow Read Parameters

@	LTH	ID	OCTET	X/b	DEF	SHADOW READ PARAMETERS
M	n+1	33				COMBINATION COMMAND EXTENT PARM (See 5.5.4)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Count
			09-0C			Data Address
			n-7:4			Count — Repeated as many
			n-3:n			Data Address — times as needed
S	n+1	34				COMBINATION RESPONSE EXTENT PARM (See 5.5.5)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Residual Count
			09-0C			Data Address
			0D-0E			Major Status
			0F- n			Substatus if any - Codes x0-xB
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	51				INFORMATION TRANSFER SIZE OVERRIDE PARAMETER (See 8.1.4.2.2)
			01-04			Generate Class 2 Interrupt
			05-08			Burst Size
M	01	52				MASTER TERMINATION PERMITTED PARAMETER (See 8.1.4.2.3)

If there are any Transfer parameters set, they shall apply to both the primary and secondary facilities.

In the event of a data check or other failure on one facility, the slave shall attempt to complete the operation from the other facility. If the command is successfully completed on the second facility, Conditional Success shall be posted as the Major Status.

If one of the facilities is not available for the operation, the slave shall complete the command to the available facility, and post Conditional Success as the Major Status.

9.6.4 Parameters. The Shadow Read Parameters shall be as shown in Table 78.

9.6.4.1 Combination Command Extent (Common) Parameter. Two parameters are required, each a source for the data to be read. See 5.5.4 and also 9.1.4.1.

9.6.4.2 Combination Response Extent (Common) Parameter. See 5.5.5 and also 9.1.4.2.

9.6.4.3 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.6.4.4 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.6.4.5 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.7.

9.6.4.6 Information Transfer Size Override Parameter. See 8.1.4.2.2.

9.6.4.7 Master Termination Permitted Parameter. See 8.1.4.2.3.

9.7 SHADOW WRITE

9.7.1 Command Packet. The command packet for this command shall be as shown in Figure 78.

9.7.2 Response Packet. The response packet for this command shall be as shown in Figure 79.

9.7.3 Description. The SHADOW WRITE command transfers data from the master to both the primary and secondary addressees starting at the location given in the respective Data Addresses. The data transfer operation shall take place first to whichever facility, primary or secondary, records a position complete.

The command executes in one of two modes depending on whether or not shadowing is on a file basis or by volume.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR	PARAMETERS 6 through n
xxxx	xxxx	36	bbbb bbbb	xx	xx		

Figure 78
Command Packet for SHADOW WRITE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 79
Response Packet for SHADOW WRITE

When executing on a file basis, the master shall specify the Count and Data Address for each Extent parameter (the PhysicalBlock sizes may be different for each facility).

When executing on a volume basis, only one Count and Data Address need be specified, since the extent on the second facility is a mirror of the first facility.

If there are any Transfer parameters set, they shall apply to both the primary and secondary facilities.

During execution, if one of the facilities should encounter a failure on one facility, the slave shall complete operation to the other facility, and post Conditional Success in the Major Status of the addressee (the cause of failure in the facility shall be identified in the Substatus of the associated Combination Response packet). It is the master's responsibility to manage the procedures necessary to recover from the loss of a single facility.

9.7.4 Parameters. The Shadow Write Parameters shall be as shown in Table 79.

9.7.4.1 Combination Command Extent (Common) Parameter. Two parameters are required, each a destination for the data to be written. See 5.5.4 and also 9.1.4.1.

9.7.4.2 Combination Response Extent (Common) Parameter. See 5.5.5 and also 9.1.4.2.

9.7.4.3 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.7.4.4 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.7.4.5 Transfer (Common) Parameter. See 5.5.13 and also 9.1.4.5.

9.7.4.6 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.6.

9.7.4.7 Information Transfer Size Override Parameter. See 8.1.4.2.2.

9.7.4.8 Master Termination Permitted Parameter. See 8.1.4.2.3.

9.8 SHADOW RESTORE

9.8.1 Command Packet. The command packet for this command shall be as shown in Figure 80.

9.8.2 Response Packet. The response packet for this command shall be as shown in Figure 81.

9.8.3 Description. The SHADOW RESTORE command is a recovery command that transfers data from the primary addressee to the master starting at the location given in the data address. The specified data shall be echoed (written) to the secondary facility, starting at the Data Address specified for same.

The effect of this is similar to that of a facility-to-facility COPY, plus transfer to the master in parallel of all data being copied. If there are any Transfer parameters set, they shall apply to the secondary facility only.

Table 79
Shadow Write Parameters

@	LTH	ID	OCTET	X/b	DEF	SHADOW WRITE PARAMETERS
M	n+1	33				COMBINATION COMMAND EXTENT PARM (See 5.5.4)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Count
			09-0C			Data Address
			n-7:4			Count — Repeated as many
			n-3:n			Data Address — times as needed
S	n+1	34				COMBINATION RESPONSE EXTENT PARM (See 5.5.5)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Residual Count
			09-0C			Data Address
			0D-0E			Major Status
			0F- n			Substatus if any - Codes x0-xB
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	51				INFORMATION TRANSFER SIZE OVERRIDE PARAMETER (See 8.1.4.2.2)
			01-04			Generate Class 2 Interrupt
			05-08			Burst Size
M	01	52				MASTER TERMINATION PERMITTED PARAMETER (See 8.1.4.2.3)

Table 80
Shadow Restore Parameters

@	LTH	ID	OCTET	X/b	DEF	SHADOW RESTORE PARAMETERS
M	n+1	33				COMBINATION COMMAND EXTENT PARM (See 5.5.4)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Count
			09-0C			Data Address
			n-7:4			Count — Repeated as many
			n-3:n			Data Address — times as needed
S	n+1	34				COMBINATION RESPONSE EXTENT PARM (See 5.5.5)
			01			Slave Address
			02			Facility Address
			03			Modifiers
			04			Reserved
			05-08			Residual Count
			09-0C			Data Address
			0D-0E			Major Status
			0F- n			Substatus if any - Codes x0-xB
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR	PARAMETERS 6 through n
xxxx	xxxx	37	bbbb bbbb	xx	xx		

Figure 80
Command Packet for SHADOW RESTORE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0011	bbbb 3210	

Figure 81
Response Packet for SHADOW RESTORE

If any failures occur on the secondary facility, the operation shall terminate with Machine Exception status reported for the failing facility.

9.8.4 Parameters. The Shadow Restore Parameters shall be as shown in Table 80.

9.8.4.1 Combination Command Extent (Common) Parameter. See 5.5.4 and also 9.1.4.1.

9.8.4.2 Combination Response Extent (Common) Parameter. See 5.5.5 and also 9.1.4.2.

9.8.4.3 Access Key (Common) Parameter. See 5.5.6 and also 9.1.4.3.

9.8.4.4 Data Address (Common) Parameter. See 5.5.11 and also 9.1.4.4.

9.8.4.5 Transfer (Common) Parameter. See 5.5.13 and also 9.1.4.5.

9.8.4.6 Partition (Common) Parameter. See 5.5.15 and also 9.1.4.6.

10. Other Transfer Commands

The data transfer commands in this section are used for specific functions other than typical read and write activity. In many situations these commands may be used to complement diagnostics. These commands by their very nature are either device-specific or vendor-specific. Refer to vendor specifications as to their implementation.

10.1 READ VERIFY

10.1.1 Command Packet. The command packet for this command shall be as shown in Figure 82.

10.1.2 Response Packet. The response packet for this command shall be as shown in Figure 83.

10.1.3 Description. The Direction modifier is not applicable to the disk.

The READ VERIFY command reads data from the addressee and verifies that the data is correct as determined by the slave's or facility's error detection/correction scheme. Data is not transferred to the master.

When used with disk, this command is used to verify that a number of blocks on the disk is formatted properly. Every identification field and PhysicalBlock on each track within the extent is read and the CRC/ECC is checked. If an error is detected, the operation is terminated and the Data Address in the Response Extent parameter identifies the block that contains the error.

10.1.3.1 High Margins Modifier. If the High Margins modifier is set and if the capability is provided within the facility, the slave shall control the circuits within the facility to operate under marginal conditions.

10.1.3.2 Volume Modifier. If the Volume modifier in the parameters is set, the slave shall verify the entire volume.

Table 81
Read Verify Parameters

@	LTH	ID	OCTET	X/b	DEF	READ VERIFY PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A				DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01- n			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	50	bbbb 7654	bbbb 3210	xx	xx	

```
                    ||||  Count  0=Octet  1=Block
                    |||  High Margins
                    ||  0=DataBlock  1=PhysicalBlock
                    |  Direction  0=Forward  1=Reverse
```

Figure 82
Command Packet for READ VERIFY

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb 7654	bbbb 3210	0001	bbbb 3210

Figure 83
Response Packet for READ VERIFY

10.1.4 Parameters. The Read Verify Parameters shall be as shown in Table 81.

10.1.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.1.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.1.4.3 Access Key (Common) Parameter. See 5.5.6.

10.1.4.4 Data Address (Common) Parameter. See 5.5.11.

10.1.4.5 Transfer (Common) Parameter. See 5.5.13.

10.1.4.6 Partition (Common) Parameter. See 5.5.15.

10.2 READ AT FIRST AVAILABLE DATA

10.2.1 Command Packet. The command packet for this command shall be as shown in Figure 84.

10.2.2 Response Packet. The response packet for this command shall be as shown in Figure 85.

10.2.3 Description. The Direction modifier is not applicable to the disk.

The READ AT FIRST AVAILABLE DATA command shall access the currently selected track and transfer data starting at the first available block. The address of the first block encountered is provided in the Read at First Data parameter appended to the Transfer Notification packet that precedes the transfer of the data.

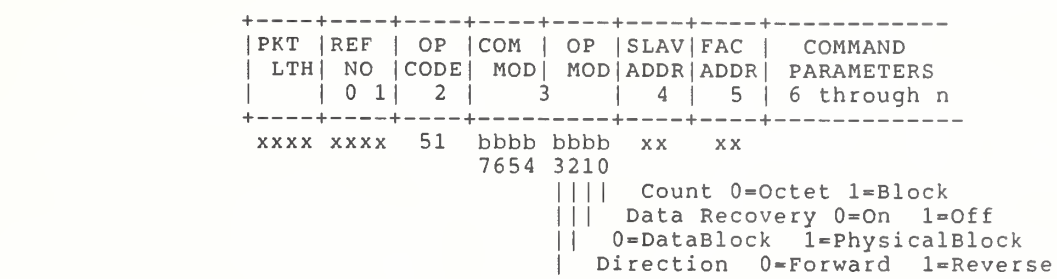


Figure 84
Command Packet for READ AT FIRST AVAILABLE DATA

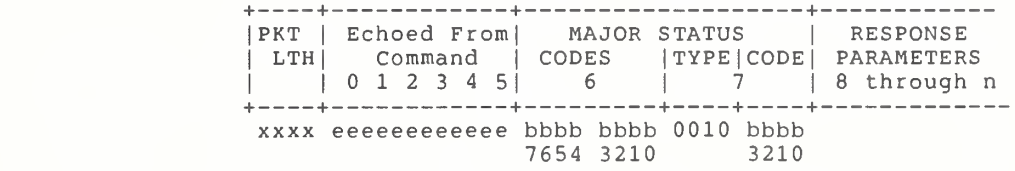


Figure 85
Response Packet for READ AT FIRST AVAILABLE DATA

Table 82
Read at First Available Data Parameters

@	LTH	ID	OCTET	X/b	DEF	READ AT FIRST AVAILABLE DATA PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	n+1	3F	01- n			STOP ON DISCONTINUITY PARAMETER (See 5.5.16 and 8.1.4.1.6)
S	09	50				READ AT FIRST DATA
			01-04			Count from Beginning
			05-08			Starting Data Address
M	n+1	51				INFORMATION TRANSFER SIZE OVERRIDE PARAMETER (See 8.1.4.2.2)
			01-04			Generate Class 2 Interrupt
			05-08			Burst Size

The data from the first block in the extent shall follow the transfer of the last block in the extent. The transfer shall continue until the Count has been exhausted.

If PhysicalBlock transfers are specified in the Command Extent parameter, the Count shall be equal to one track of data. The facility shall not perform a track or cylinder change during the transfer.

If DataBlock transfers are specified, the addressee shall transfer all blocks within the extent, from point of beginning transfer to end of extent, then beginning of extent to the block prior to the first transferred.

If the Block Size parameter is supplied, the slave shall use the specified block size only for the duration of the command execution.

The master does not know from which address the first data will be transferred, so the slave shall append the Read at First Data parameter to the Transfer Notification Response (which is required for execution of this command), and also to the response packet.

The master is responsible for making the necessary corrections (if any) in the sequence of data in order to compensate for its improper order. READ AT FIRST AVAILABLE DATA is intended to provide improved performance;but it must be recognized that due to the addressee's and the master's overhead in order to execute this command, some operations may perform slower than if the data was read normally. For this reason, the extent should normally be restricted to a physical track or cylinder.

10.2.4 Parameters. The Read At First Available Data Parameters shall be as shown in Table 82.

10.2.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.2.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.2.4.3 Access Key (Common) Parameter. See 5.5.6.

10.2.4.4 Data Address (Common) Parameter. See 5.5.11.

10.2.4.5 Partition (Common) Parameter. See 5.5.15.

10.2.4.6 Stop On Discontinuity Parameter. See 5.5.16 and also 8.1.4.1.6.

10.2.4.7 Read at First Data Parameter

(1) *Count from Beginning.* This value specifies the displacement into the extent in which the transfer begins.

(2) *Starting Data Address.* This field identifies the address within the extent from which the slave shall begin transferring data.

10.2.4.8 Information Transfer Size Override Parameter. See 8.1.4.2.2.

10.3 READ FROM BUFFER

10.3.1 Command Packet. The command packet for this command shall be as shown in Figure 86.

10.3.2 Response Packet. The response packet for this command shall be as shown in Figure 87.

10.3.3 Description. The READ FROM BUFFER command transfers the contents of the addressee's buffer to the master beginning at the octet offset contained in the Data Address field of the Command Extent parameter. The Count specifies the number of octets that are to be transferred to the master.

READ FROM BUFFER may be used in conjunction with WRITE TO BUFFER to test the addressee's data buffer. To ensure that the contents of the buffer from the WRITE TO BUFFER command are the same as can be read, the master has to Order their execution to be certain that a command-queuing slave did not use the buffer when overlapping command executions.

The addressee shall transfer the specified number of octets from the buffer to the master.

The buffer to be used is identified in the Buffer Address parameter. One of the following buffers may be used:

(1) *The "Generic" Buffer.* This is the buffer that the slave normally makes visible to the master and may be in either the slave or the addressee.

(2) *The Slave Buffer.* This buffer is always contained in the slave.

(3) *The Facility Buffer.* This buffer is always in the facility.

10.3.4 Parameters. The Read From Buffer Parameters shall be as shown in Table 83.

10.3.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.3.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.3.4.3 Access Key (Common) Parameter. See 5.5.6.

10.3.4.4 Data Address (Common) Parameter. See 5.5.11.

10.3.4.5 Partition (Common) Parameter. See 5.5.15.

10.3.4.6 Buffer Address Parameter. The first octet identifies buffer that is to be used.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	52	bbbb bbbb		xx	xx	

Figure 86
Command Packet for READ FROM BUFFER

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 7	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 87
Response Packet for READ FROM BUFFER

Table 83
Read From Buffer Parameters

@	LTH	ID	OCTET	X/b	DEF	READ FROM BUFFER PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	03	50	01			BUFFER ADDRESS PARAMETER
				7		Generic *
				6		Slave *
				5		Facility *
				4-0		Reserved
			02	7		Port Command Stack *
				6		Slave Command Stack *
				5		Slave Data Buffer *
				4-0		Reserved

* Mutually exclusive.

Table 84
Read Facility Data To Buffer Parameters

@	LTH	ID	OCTET	X/b	DEF	READ FACILITY DATA TO BUFFER PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	03	50	01-02			BUFFER ADDRESS PARAMETER (See 10.3.4.6)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx	xxxx	53	bbbb 7654	bbbb 3210	xx	xx	

3210: |||| Count 0=Octet 1=Block; ||| Data Recovery 0=On 1=Off; || 0=DataBlock 1=PhysicalBlock; | Direction 0=Forward 1=Reverse

Figure 88
Command Packet for READ FACILITY DATA TO BUFFER

(1) *Bit 7 = 1.* The buffer utilized by the slave is to be used.

(2) *Bit 6 = 1.* The slave buffer is to be used. If none is present, this parameter shall be rejected.

(3) *Bit 5 = 1.* The facility buffer is to be used. If none is present, this parameter shall be rejected.

The second octet may be used to identify the buffers within the slave that are to be addressed (if any). It is possible for a slave to use several buffers in an implementation.

10.4 READ FACILITY DATA TO BUFFER

10.4.1 Command Packet. The command packet for this command shall be as shown in Figure 88.

10.4.2 Response Packet. The response packet for this command shall be as shown in Figure 89.

10.4.3 Description. The Direction modifier is not applicable to the disk.

The READ FACILITY DATA TO BUFFER command is similar to a READ command except that data is transferred to a specified buffer rather than to the master. The transfer size shall be less than or equal to the length of the destination buffer.

10.4.4 Parameters. The Read Facility Data To Buffer Parameters shall be as shown in Table 84.

10.4.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.4.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.4.4.3 Access Key (Common) Parameter. See 5.5.6.

10.4.4.4 Data Address (Common) Parameter. See 5.5.11.

10.4.4.5 Transfer (Common) Parameter. See 5.5.15.

10.4.4.6 Buffer Address Parameter. See 10.3.4.6.

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
	xxxx eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 89
Response Packet for READ FACILITY DATA TO BUFFER

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
	xxxx xxxx	54	bbbb bbbb 7654 3210		xx	xx	

|||| Count 0=Octet 1=Block
||| 0=ECC 1=Syndrome
||
| Direction 0=Forward 1=Reverse

Figure 90
Command Packet for READ PHYSICAL DATA AND ECC

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
	xxxx eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 91
Response Packet for READ PHYSICAL DATA AND ECC

10.5 READ PHYSICAL DATA AND ECC

10.5.1 Command Packet. The command packet for this command shall be as shown in Figure 90.

10.5.2 Response Packet. The response packet for this command shall be as shown in Figure 91.

10.5.3 Description. The Direction modifier is not applicable to the disk.

The READ PHYSICAL DATA AND ECC command reads data beginning at the PhysicalBlock specified by the Data Address field in the Command Extent parameter. The error detection/correction information read with the data is appended and transfered with the data. The data is checked for errors and, if errors are detected, Substatus is reported in the response packet.

The ECC information returned with the data may be either the recorded ECC information or may be the syndrome, as specified by the modifier octet.

Operation of this command shall be limited to the PhysicalBlock protected by the ECC.

10.5.4 Parameters. The Read Physical Data And ECC Parameters shall be as shown in Table 85.

10.5.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.5.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.5.4.3 Access Key (Common) Parameter. See 5.5.6.

10.5.4.4 Data Address (Common) Parameter. See 5.5.11.

10.5.4.5 Partition (Common) Parameter. See 5.5.15.

Table 85
Read Physical Data And ECC Parameters

@	LTH	ID	OCTET	X/b	DEF	READ PHYSICAL DATA AND ECC PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx xxxx		55	bbbb bbbb 7654 3210		xx	xx	

```
                    ||||  Count  0=Octet  1=Block
                    |||  Data Recovery 0=On 1=Off
                    ||
                    |  Direction  0=Forward  1=Reverse
```

Figure 92
Command Packet for READ PHYSICAL HEADER

10.6 READ PHYSICAL HEADER

10.6.1 Command Packet. The command packet for this command shall be as shown in Figure 92.

10.6.2 Response Packet. The response packet shall be as shown in Figure 93.

10.6.3 Description. The Direction modifier is not applicable to the disk.

The READ PHYSICAL HEADER command reads the identification fields that are transferred as data starting at the Physical-Block specified by the Data Address field in the Command Extent parameter for the extent defined by the Count. The Count shall not be greater than the number of PhysicalBlocks on a track.

10.6.4 Parameters. The Read Physical Header Parameters shall be as shown in Table 86.

10.6.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.6.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.6.4.3 Access Key (Common) Parameter. See 5.5.6.

10.6.4.4 Data Address (Common) Parameter. See 5.5.11.

10.6.4.5 Partition (Common) Parameter. See 5.5.15.

10.7 READ IPL

10.7.1 Command Packet. The command packet for this command shall be as shown in Figure 94.

10.7.2 Response Packet. The response packet for this command shall be as shown in Figure 95.

Table 86
Read Physical Header Parameters

@	LTH	ID	OCTET	X/b	DEF	READ PHYSICAL HEADER PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb	0001	bbbb	
		7654 3210		3210	

Figure 93
Response Packet for READ PHYSICAL HEADER

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	56	bbbb	bbbb	xx	xx	

Figure 94
Command Packet for READ IPL

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb	0001	bbbb	
		7654 3210		3210	

Figure 95
Response Packet for READ IPL

Table 87
Read IPL Parameters

@	LTH	ID	OCTET	X/b	DEF	READ IPL PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx	xxxx	58	bbbb (7654)	bbbb (3210)	xx	xx	

|||| 0=ECC 1=Syndrome
||
| Direction 0=Forward 1=Reverse

Figure 96
Command Packet for READ PHYSICAL HEADER AND ECC

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	STATUS TYPE 7	STATUS CODE 7	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb (7654 3210)	0001	bbbb (3210)	

Figure 97
Response Packet for READ PHYSICAL HEADER AND ECC

10.7.3 Description. The READ IPL (Initial Program Load) command causes the first block of IPL data to be transferred to the master. Typically, receipt of the READ IPL command shall cause the slave to access the first block of the IPL area on the addressed facility. When the access is complete, the addressee shall verify correct access position and shall transfer one or more blocks of data to the master.

10.7.4 Parameters. The Read IPL Parameters shall be as shown in Table 87.

10.7.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.7.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.7.4.3 Access Key (Common) Parameter. See 5.5.6.

10.7.4.4 Data Address (Common) Parameter. See 5.5.11.

10.7.4.5 Partition (Common) Parameter. See 5.5.15.

10.8 READ PHYSICAL HEADER AND ECC

10.8.1 Command Packet. The command packet for this command shall be as shown in Figure 96.

10.8.2 Response Packet. The response packet for this command shall be as shown in Figure 97.

Table 88
Read Physical Header And ECC Parameters

@	LTH	ID	OCTET	X/b	DEF	READ PHYSICAL HEADER AND ECC PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx	xxxx	62	bbbb	bbbb	xx	xx	

Figure 98
Command Packet for WRITE TO BUFFER

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	STATUS TYPE 7	STATUS CODE 7	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 99
Response Packet for WRITE TO BUFFER

10.8.3 Description. The Direction modifier is not applicable to the disk.

On disks the READ PHYSICAL HEADER command causes the identification fields to be read from the PhysicalBlock specified by the Data Address field in the Command Extent parameter. The Count shall not be greater than the number of PhysicalBlocks on a track.

The ECC (or may be CRC) information returned with the header may be either the recorded ECC information or may be the syndrome, as specified by the modifier octet.

10.8.4 Parameters. The Read Physical Header And ECC Parameters shall be as shown in Table 88.

10.8.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.8.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.8.4.3 Access Key (Common) Parameter. See 5.5.6.

10.8.4.4 Data Address (Common) Parameter. See 5.5.11.

10.8.4.5 Partition (Common) Parameter. See 5.5.15.

10.9 WRITE TO BUFFER

10.9.1 Command Packet. The command packet for this command shall be as shown in Figure 98.

10.9.2 Response Packet. The response packet for this command shall be as shown in Figure 99.

Table 89
Write to Buffer Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE TO BUFFER PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	03	50	01			BUFFER ADDRESS PARAMETER
				7		Generic *
				6		Slave *
				5		Facility *
				4-0		Reserved
			02	7		Port Command Stack *
				6		Slave Command Stack *
				5		Slave Data Buffer *
				4-0		Reserved

* Mutually exclusive.

10.9.3 Description. The WRITE TO BUFFER command transfers data from the master to the designated addressee buffer beginning at the octet offset contained in the Data Address field of the Command Extent parameter. The Count specifies the number of octets that are to be transferred, which shall be equal to or less than the size of the destination buffer.

The WRITE TO BUFFER command may be used in conjunction with the READ FROM BUFFER command to test the addressee's data buffer. The addressee shall transfer the specifed number of octets from the master to its buffer.

10.9.4 Parameters. The Write To Buffer Parameters shall be as shown in Table 89.

10.9.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.9.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.9.4.3 Access Key (Common) Parameter. See 5.5.6.

10.9.4.4 Data Address (Common) Parameter. See 5.5.11.

10.9.4.5 Partition (Common) Parameter. See 5.5.15.

10.9.4.6 Buffer Address Parameter. See 10.3.4.6.

10.10 WRITE BUFFER TO FACILITY

10.10.1 Command Packet. The command packet for this command shall be as shown in Figure 100.

10.10.2 Response Packet. The response packet for this command shall be as shown in Figure 101.

10.10.3 Description. The Direction modifier is not applicable to the disk.

The WRITE BUFFER TO FACILITY command is similar to a WRITE command except that the data is written from the addressee's buffer. Data shall not be transferred from the master. The Count shall be less than or equal to the length of the specified buffer.

The source buffer for the write is defined by the Buffer Address parameter and is identical to the buffer addressing options available in the READ FACILITY DATA TO BUFFER command.

10.10.4 Parameters. The Write Buffer To Facility Parameters shall be as shown in Table 90.

10.10.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.10.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.10.4.3 Access Key (Common) Parameter. See 5.5.6.

10.10.4.4 Data Address (Common) Parameter. See 5.5.11.

10.10.4.5 Partition (Common) Parameter. See 5.5.15.

10.10.4.6 Buffer Address Parameter. See 10.3.4.5.

```
+----+----+----+----+----+----+----+-----------
|PKT |REF | OP |COM | OP |SLAV|FAC |  COMMAND
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR| PARAMETERS
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+------------
 xxxx xxxx  63  bbbb bbbb  xx   xx
                7654 3210
                     ||||  Count  0=Octet  1=Block
                     |||
                     ||  0=DataBlock  1=PhysicalBlock
                     |  Direction  0=Forward  1=Reverse
```

Figure 100
Command Packet for WRITE BUFFER TO FACILITY

```
+----+------------+-------------------+------------
|PKT | Echoed From|    MAJOR STATUS    |  RESPONSE
| LTH|   Command  | CODES   |TYPE|CODE| PARAMETERS
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+------------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 101
Response Packet for WRITE BUFFER TO FACILITY

Table 90
Write Buffer To Faility Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE BUFFER TO FACILITY PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	02	3C	01			TRANSFER PARAMETERS (See 5.5.13)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)
M	02	50	01			BUFFER ADDRESS PARAMETER (See 10.3.4.5)
				7		Generic *
				6		Slave *
				5		Facility *
				4-0		Reserved

* Mutually exclusive.

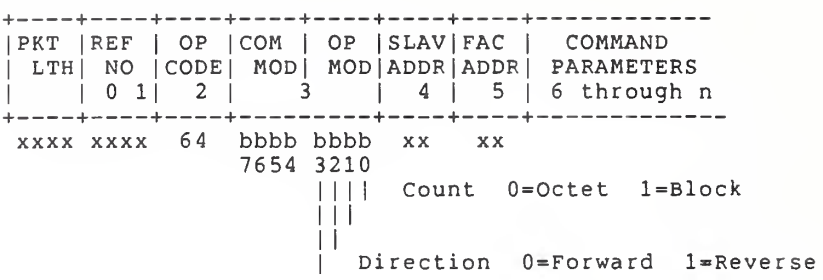


Figure 102
Command Packet for WRITE PHYSICAL DATA AND ECC

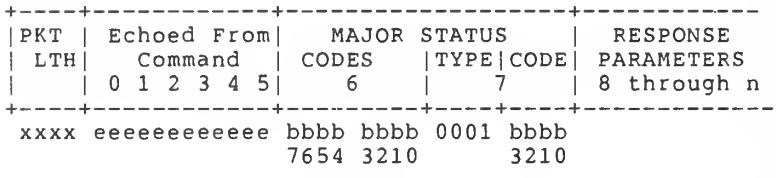


Figure 103
Response Packet for WRITE PHYSICAL DATA AND ECC

10.11 WRITE PHYSICAL DATA AND ECC

10.11.1 Command Packet. The command packet for this command shall be as shown in Figure 102.

10.11.2 Response Packet. The response packet for this command shall be as shown in Figure 103.

10.11.3 Description. The Direction modifier is not applicable to the disk.

The WRITE PHYSICAL DATA AND ECC command writes data and associated error detection/correction code information on the facility beginning at the PhysicalBlock specified by the Data Address field in the Command Extent parameter.

10.11.4 Parameters. The Write Physical Data And ECC Parameters shall be as shown in Table 91.

10.11.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.11.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.11.4.3 Access Key (Common) Parameter. See 5.5.6.

10.11.4.4 Data Address (Common) Parameter. See 5.5.11.

10.11.4.5 Partition (Common) Parameter. See 5.5.15.

10.12 WRITE PHYSICAL HEADER

10.12.1 Command Packet. The command packet for this command shall be as shown in Figure 104.

10.12.2 Response Packet. The response packet for this command shall be as shown in Figure 105.

10.12.3 Description. The Direction modifier is not applicable to the disk.

The WRITE PHYSICAL HEADER command writes the identification fields that are transferred as data starting at the PhysicalBlock specified by the Data Address field in the Command Extent parameter for the extent defined by the Count.

10.12.4 Parameters. The Write Physical Header Parameters shall be as shown in Table 92.

Table 91
Write Physical Data and ECC Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE PHYSICAL DATA AND ECC PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx xxxx		65	bbbb bbbb 7654 3210		xx	xx	

```
                    ||||  Count 0=Octet 1=Block
                    |||
                    ||
                    |  Direction 0=Forward 1=Reverse
```

Figure 104
Command Packet for WRITE PHYSICAL HEADER

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 105
Response Packet for WRITE PHYSICAL HEADER

Table 92
Write Physical Header Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE PHYSICAL HEADER PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx xxxx		66	bbbb bbbb 7654 3210		xx	xx	

```
                    ||||    Bits 0-2 have encoded meaning of
                    |||     x'0'=Load      x'1'=Reload
                    ||      x'2'=Report    x'4'=Store
```

Figure 106
Command Packet for LOAD SLAVE IML

10.12.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.12.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.12.4.3 Access Key (Common) Parameter. See 5.5.6.

10.12.4.4 Data Address (Common) Parameter. See 5.5.11.

10.12.4.5 Partition (Common) Parameter. See 5.5.15.

10.13 LOAD SLAVE IML

10.13.1 Command Packet. The command packet for this command shall be as shown in Figure 106.

10.13.2 Response Packet. The response packet for this command shall be as shown in Figure 107.

10.13.3 Description. LOAD SLAVE IML causes the slave to act upon its Initial Microcode Load according to the opcode modifiers. The slave shall perform the requested operation on/to its microcode area, then perform any necessary checks on the data as specified in the product specification.

On successful Load (x'0') or Reload (x'1'), the slave shall become ready for use with the new microcode set in place.

The opcode modifier defines whether the master wishes to Load (x'0'), Reload (x'1'), Report (x'2'), or Store (x'4') the addressee IML area. The modifiers are mutually exclusive; i.e., only one action may be specified by the command modifier.

Load causes the slave to transfer an Initial Microcode Load (IML) from the master. The slave shall load the data (as specified in the product specification) into its microcode area, perform its checks on the data, and become ready for use. Load IML will not update the IML partition on any attached media. The Load may only be addressed to the slave.

Execution of Reload, Report, or Store requires one of the following:

(1) The OPERATING MODE command chained to the LOAD SLAVE IML command

(2) The Partition parameter appended to the command packet

Reload allows the master to request that the addressee read its Initial Microcode from the appropriate area in the IML partition.

Report requests the addressee to respond with an information transfer containing the contents of the addressee IML area. If the addressee is the slave, the source of the data shall be the RAM/ROM area in which the current microcode is executing. If the addressee is a facility, the data represents the IML partition of the attached facility.

Store causes the addressee to load its currently executing microcode into the appropriate area in the IML partition of a facility. This command allows updates of the slave's microcode to be written to the facility.

NOTE: The Command Completion response sent to the master by the slave for this command should be transferred only after the newly loaded microcode has been successfully initialized and is operational.

10.13.4 Parameters. The load slave IML parameters shall be as shown in Table 93.

10.13.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.13.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.13.4.3 Partition (Common) Parameter. See 5.5.15.

10.14 ERASE

10.14.1 Command Packet. The command packet for this command shall be as shown in Figure 108.

10.14.2 Response Packet. The response packet for this command shall be as shown in Figure 109.

Table 93
Load Slave IML Parameters

@	LTH	ID	OCTET	X/b	DEF	LOAD SLAVE IML PARAMETERS
M	n+1	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
M	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Count
			05-08			Data Address
M	n+1	3E				PARTITION PARAMETER (See 5.5.15)

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 107
Response Packet for LOAD SLAVE IML

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx	xxxx	67	bbbb bbbb 7654 3210	xx	xx		

4 = Count 0=Octet 1=Block
3 = 0=DataBlock 1=PhysicalBlock
2 = Direction 0=Forward 1=Reverse

Figure 108
Command Packet for ERASE

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 109
Response Packet for ERASE

Table 94
Erase Parameters

@	LTH	ID	OCTET	X/b	DEF	ERASE PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

10.14.3 Description. The Direction modifier is not applicable to the disk.

The ERASE command writes through all addressable recording spaces of the addresses specified in the command packet. The data and identification fields in the recording spaces operated on by the ERASE command shall be unformatted and unrecognizable as a result of the command.

10.14.4 Parameters. The Erase Parameters shall be as shown in Table 94.

10.14.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.14.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.14.4.3 Access Key (Common) Parameter. See 5.5.6.

10.14.4.4 Data Address (Common) Parameter. See 5.5.11.

10.14.4.5 Partition (Common) Parameter. See 5.5.15.

10.15 WRITE PHYSICAL HEADER AND ECC

10.15.1 Command Packet. The command packet for this command shall be as shown in Figure 110.

10.15.2 Response Packet. The response packet for this command shall be as shown in Figure 111.

10.15.3 Description. The Direction modifier is not applicable to the disk.

The WRITE PHYSICAL HEADER command writes the identification fields starting at the PhysicalBlock specified by the Data Address field in the Command Extent parameter for the extent defined by the Count.

10.15.4 Parameters. The Write Physical Header And ECC Parameters shall be as shown in Table 95.

10.15.4.1 Command Extent (Common) Parameter. See 5.5.2.

10.15.4.2 Response Extent (Common) Parameter. See 5.5.3.

10.15.4.3 Access Key (Common) Parameter. See 5.5.6.

10.15.4.4 Data Address (Common) Parameter. See 5.5.11.

10.15.4.5 Partition (Common) Parameter. See 5.5.1.

```
+----+----+----+----+----+----+----+-----------
|PKT |REF | OP |COM | OP |SLAV|FAC |  COMMAND
| LTH| NO |CODE| MOD| MOD|ADDR|ADDR| PARAMETERS
|    | 0 1|  2 |    3    |  4 |  5 | 6 through n
+----+----+----+---------+----+----+-------------
 xxxx xxxx  68  bbbb bbbb  xx   xx
                7654 3210
                     ||||  Count 0=Octet 1=Block
                     |||
                     ||
                     |  Direction 0=Forward 1=Reverse
```

Figure 110
Command Packet for WRITE PHYSICAL HEADER AND ECC

```
+----+------------+--------------------+------------
|PKT | Echoed From|    MAJOR STATUS    |  RESPONSE
| LTH|   Command  | CODES   |TYPE|CODE| PARAMETERS
|    | 0 1 2 3 4 5|    6    |    7    | 8 through n
+----+------------+---------+----+----+-------------
 xxxx eeeeeeeeeeee bbbb bbbb 0001 bbbb
                   7654 3210      3210
```

Figure 111
Response Packet for WRITE PHYSICAL HEADER AND ECC

Table 95
Write Physical Header And ECC Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE PHYSICAL HEADER AND ECC PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	05	35	01-04			ACCESS KEY PARAMETER (See 5.5.6)
B	n+1	3A	01- n			DATA ADDRESS PARAMETER (See 5.5.11)
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 7 through n
xxxx xxxx		80	bbbb bbbb 7654 3210		xx	xx	

|||| Count 0=Octet 1=Block

Figure 112
Command Packet for PERFORM SLAVE DIAGNOSTICS

11. Diagnostic Commands

The commands in this section are the maintenance and diagnostic commands of the Logical Interface. These commands by their very nature are either device-specific or vendor-specific. Refer to vendor specifications as to their implementation.

The commands in this section may require that parameters be transferred as data (because they are too big to transfer as parameters). (See 5.2.4.) The master shall set the Count field of the Command Extents parameter to the limit to be transferred.

11.1 PERFORM SLAVE DIAGNOSTICS

11.1.1 Command Packet. The command packet for this command shall be as shown in Figure 112.

11.1.2 Response Packet. The response packet for this command shall be as shown in Figure 113.

11.1.3 Description. The PERFORM SLAVE DIAGNOSTICS command causes the specified self-contained slave diagnostic to be executed. Results from the slave diagnostic are slave unique, and may be returned to the master as data.

11.1.4 Parameters. The Perform Slave Diagnostics Parameters shall be as shown in Table 96.

11.1.4.1 Command Extent (Common) Parameter. See 5.5.2.

11.1.4.2 Response Extent (Common) Parameter. See 5.5.3

11.1.4.3 Diagnostic Number Parameter. This parameter supplies a number that has a specific meaning, as one of a list of documented diagnostics identified in the vendor documentation.

11.1.4.4 Diagnostic Function List Parameter. This parameter supplies a list of functions to be performed as documented by the vendor.

11.1.4.5 Diagnostic Action Code Parameter. This parameter supplies an action code and a list of parameters associated with it, as documented by the vendor.

11.1.4.6 Vendor-Unique Parameter. The field or fields, if any, in this parameter shall be specified in the vendor's documentation.

11.1.4.7 Request Parm Parameter. See 6.3.13.1.

11.2 PERFORM FACILITY DIAGNOSTICS

11.2.1 Command Packet. The command packet for this command shall be as shown in Figure 114.

11.2.2 Response Packet. The response packet for this command shall be as shown in Figure 115.

11.2.3 Description. The PERFORM FACILITY DIAGNOSTICS command shall cause the specified self-contained facility diagnostic to be executed. Results from the facility diagnostic are facility-unique and may be returned to the master as data.

This command is valid only if the vendor specification of the device or devices attached to a slave are capable of executing diagnostics that provide information to the master.

11.2.4 Parameters. The Perform Facility Diagnostic Parameters shall be as shown in Table 97.

11.2.4.1 Command Extent (Common) Parameter. See 5.5.2.

11.2.4.2 Response Extent (Common) Parameter. See 5.5.3.

11.2.4.3 Diagnostic Number Parameter. See 11.1.4.3.

11.2.4.4 Diagnostic Function List Parameter. See 11.1.4.4.

11.2.4.5 Diagnostic Action Code Parameter. See 11.1.4.5.

11.2.4.6 Vendor-Unique Parameter. See 11.1.4.6.

11.2.4.7 Request Parm Parameter. See 6.3.4.13.1.

Table 96
Perform Slave Diagnostics Parameters

@	LTH	ID	OCTET	X/b	DEF	PERFORM SLAVE DIAGNOSTIC PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
M	03	50	01-02			DIAGNOSTIC NUMBER PARAMETER (vendor specific)
M	n+1	51	01- n			DIAGNOSTIC FUNCTION LIST PARAMETER
						(vendor specific)
M	n+1	52	01-02			DIAGNOSTIC ACTION CODE PARAMETER
						(vendor specific)
			03- n			DIAGNOSTIC PARAMETERS (vendor specific)
M	n+1	53	01- n			VENDOR-UNIQUE PARAMETER
M	n+1	6C				REQUEST PARM PARAMETER
			01	7		Parameters as Data *
				6		Parameters in Response *
				5		Reserved
				4		Naked Parameters as Data *
				3-0		Reserved
			02			Parameter ID Repeated as many
			n			Parameter ID times as needed
						* mutually exclusive parameters

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 8 through n	RESPONSE PARAMETERS
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 113
Response Packet for PERFORM SLAVE DIAGNOSTICS

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6 through n	COMMAND PARAMETERS
xxxx	xxxx	81	bbbb bbbb 7654 3210	xx	xx		

|||| Count 0=Octet 1=Block

Figure 114
Command Packet for PERFORM FACILITY DIAGNOSTICS

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 8 through n	RESPONSE PARAMETERS
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 115
Response Packet for PERFORM FACILITY DIAGNOSTICS

Table 97
Perform Facility Diagnostics Parameters

@	LTH	ID	OCTET	X/b	DEF	PERFORM FACILITY DIAGNOSTIC PARAMETERS	
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)	
			01-04			Count	
			05-08			Data Address	
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)	
			01-04			Residual Count	
			05-08			Data Address	
M	03	50	01-02			DIAGNOSTIC NUMBER PARAMETER (vendor specific)	
M	n+1	51	01- n			DIAGNOSTIC FUNCTION LIST PARAMETER (vendor specific)	
M	n+1	52	01-02			DIAGNOSTIC ACTION CODE PARAMETER (vendor specific)	
			03- n			DIAGNOSTIC PARAMETERS (vendor specific)	
M	n+1	53	01- n			VENDOR-UNIQUE PARAMETER	
M	n+1	6C				REQUEST PARM PARAMETER	
			01	7		Parameters as Data	*
				6		Parameters in Response	*
				5		Reserved	
				4		Naked Parameters as Data	*
				3-0		Reserved	
			02			Parameter ID	Repeated as many
			n			Parameter ID	times as needed

* Mutually exclusive parameters.

11.3 READ DEFECT LIST

11.3.1 Command Packet. The command packet for this command shall be as shown in Figure 116.

11.3.2 Response Packet. The response packet for this command shall be as shown in Figure 117.

11.3.3 Description. The READ DEFECT LIST command transfers the Working Permanent, Working Temporary, Suspect Permanent, or Suspect Temporary media defect list as specified by the modifier octet, for the partition defined in the Partition parameter (if any), or by a preceding OPERATING MODE command (if any) in a Chain, Sequence, or Order.

Note that the media defect list may be returned as data, although it consists of the parameter list containing the requested information. Media defect information is kept in a variety of ways by different vendors, and the parameter format provides a common method of communicating the information.

(1) The Permanent or Temporary modifier identifies whether the permanent or temporary list is to be used

(2) The Working or Suspect modifier identifies whether the working or suspect list is to be used

(3) The Sequential modifier indicates that the requested defect list entries are to be returned in order of ascending address. The Chronological modifier indicates that the requested defect list entries are to be returned in the order in which media replacements occurred.

Chronological defect lists have several special characteristics. For uses of the ALLOCATE RESTORE command that must operate on defects in reverse chronological order, this list provides that order. If a REALLOCATE command is being used to restore a slave's Working set of defects, the chronological list provides the order of reallocations that will restore the current condition. Duplicate entries may appear in chronological defect lists to indicate multiple defects in the same block over a period of time.

11.3.4 Read Defect List Parameters

11.3.4.1 Parameters 31-57. These parameters shall be as shown in Table 98.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	82	bbbb 7654	bbbb 3210	xx	xx	

```
                    ||||  Count  0=Octet  1=Block
                    |||  0=Permanent   1=Temporary
                    ||  0=Working     1=Suspect
                    |  0=Sequential  1=Chronological
```

Figure 116
Command Packet for READ DEFECT LIST

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb 7654	bbbb 3210	0001	bbbb 3210

Figure 117
Response Packet for READ DEFECT LIST

Table 98
Read Defect List Parameters 31-57

@	LTH	ID	OCTET	X/b	DEF	READ DEFECT LIST PARAMETERS	
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)	
			01-04			Count	
			05-08			Data Address	
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)	
			01-04			Residual Count	
			05-08			Data Address	
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)	
S	n+1	55				DEFECTIVE DATABLOCK PARAMETER	
			01-02			Size of DataBlock Address Fields	
			03-			Address of Defective DataBlock	(first)
			n- :n			Address of Defective DataBlock	(last)
S	n+1	56				TRACK DEFECTS LIST PARAMETER	
			01-04			Cylinder	(first)
			05-06			Track	
			07-0A			Octet Offset into track	
			0B-0C			Length of defect (in bits)	
			n-B:8			Cylinder	(last)
			n-7:6			Track	
			n-5:2			Octet Offset into track	
			n-1:n			Length of defect (in bits)	
S	n+1	57				SECTOR DEFECTS LIST PARAMETER	
			01-04			Cylinder	(first)
			05-06			Track	
			07-08			No of Sector after Index	
			09-0A			Octet Offset within Sector	
			0B-0C			Length of defect (in bits)	
			n-B:8			Cylinder	(last)
			n-7:6			Track	
			n-5:4			No of Sector after Index	
			n-3:2			Octet Offset within Sector	
			n-1:n			Length of defect (in bits)	

11.3.4.1.1 Command Extent (Common) Parameter. This parameter identifies the count of data to be transferred and the offset within the list at which transfer is to begin (see also 5.5.2).

11.3.4.1.2 Response Extent (Common) Parameter. See 5.5.3.

11.3.4.1.3 Partition (Common) Parameter. See 5.5.15.

11.3.4.1.4 Defective DataBlock Parameter. This parameter may be returned as data to indicate DataBlocks that currently contain defective media, or DataBlocks that have been affected by defect reallocation. This depends on which list is read, and vendor-specific defect management algorithms (see also 9.4.4.2.2).

11.3.4.1.5 Track Defects List Parameter. See 8.7.6.3.1.

11.3.4.1.6 Sector Defects List Parameter. See 8.7.6.3.2.

11.3.4.2 Parameters 58-6D. These parameters shall be as shown in Table 99.

11.3.4.2.1 Variable Size Sectors Parameter. This parameter immediately precedes the first Track Defects List parameter. The first value defines the Number of Cells at the Index Gap, and the second defines the number of octets in a regular cell. The third field contains the number of cells reserved per track for skipping flaws.

11.3.4.2.2 Cell Defects List Parameter. See 8.7.6.4.1.

11.3.4.2.3 Request Parm Parameter. See 6.3.4.13.1.

11.3.4.2.4 Parm Length Parameter. See 6.3.4.13.2.

11.4 WRITE DEFECT LIST

11.4.1 Command Packet. The command packet for this command shall be as shown in Figure 118.

11.4.2 Response Packet. The response packet for this command shall be as shown in Figure 119.

11.4.3 Description. The WRITE DEFECT LIST command transfers addresses of Permanent or Temporary defects, as determined by the modifier byte, to the Suspect Permanent or Suspect Temporary defect list for the partition defined in the Partition parameter (if any), or by a preceding OPERATING MODE command (if any) in a Chain, Sequence, or Order.

WRITE DEFECT LIST does not cause the slave to begin any reallocation of data.

The location and format of the defect list are slave or facility dependent. Note that the media defect list may be transferred as data. The Request Parms parameter shall identify if the list is to be transferred as data, and if it is to be a general transfer or a specific transfer, containing only the contents of the parameter.

Entries in the Suspect defect list do not cause media replacement until the next FORMAT or REALLOCATE command.

(1) The Permanent or Temporary modifier identifies the Suspect list to be used.

(2) The Clear Defect List modifier is used to clear the entire list of all entries. It allows the master to force the slave to reference the manufacturer's initial list of flaws when a succeeding FORMAT command is issued.

(3) The Create or Append modifier is used to break up a long parameter list into multiple smaller ones that can be transferred to the slave. The Create clears any existing information in the addressee's Suspect defect list as identified by the modifier.

If the Create modifier is set, the old Suspect list is replaced by the list transferred with this command. Otherwise, the transferred defect list shall be appended to the existing Suspect defect list.

To supply defect information for areas other than the Data Partition, it is necessary to precede this command in a Chain, Sequence, or Order with an OPERATING MODE command that identifies the required partition.

It is recommended that the defect lists be kept with the media to which they are applicable. In the case of removable media, this implies that the lists need to be recorded on the media.

11.4.4 Write Defect List Parameters

11.4.4.1 Parameters 31-57. These parameters shall be as shown in Table 100.

11.4.4.1.1 Command Extent (Common) Parameter. See 5.5.2.

11.4.4.1.2 Response Extent (Common) Parameter. See 5.5.3.

11.4.4.1.3 Partition (Common) Parameter. See 5.5.15.

11.4.4.1.4 Defective DataBlock Parameter. This parameter contains a list of DataBlock addresses that are to be placed in a defect list (see also 9.4.4.2.2).

11.4.4.1.5 Track Defects List Parameter. See 8.7.6.3.1.

Table 99
Read Defect List Parameters 58-6D

@	LTH	ID	OCTET	X/b	DEF	READ DEFECT LIST PARAMETERS
S	09	58				VARIABLE SIZE SECTORS PARAMETER
			01-02			Size of Cells at Index Gap
			03-04			Cell Size
			05-06			No of Cells Reserved per Track to Skip Flaws
			07-08			No of Cells to Skip per Flaw
S	n+1	5A				CELL DEFECTS LIST PARAMETER
			01-04			Cylinder (first)
			05-06			Track
			07-08			No of Defective Cells
			09-0C			Offset of Defective Cell #1
			0D- m			Offset of Defective Cell #2 through end
			n -			Cylinder (last)
			n -			Track
			n -			No of Defective Cells
			n -			Offset of Defective Cell #1
			n -			Offset of Defective Cell #2 through end
M	n+1	6C				REQUEST PARM PARAMETER
			01	7		Parameters as Data *
				6		Parameters in Response *
				5		Length *
				4		Naked Parameters as Data *
				3-0		Reserved
			02			Parameter ID Repeated as many
			n			Parameter ID times as needed
S	05	6D				PARM LENGTH PARAMETER
			01-04			Length of Parameter List

* Mutually exclusive parameters.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6 through n	COMMAND PARAMETERS
xxxx	xxxx	83	bbbb bbbb 7654 3210	xx	xx		

|||| Count 0=Octet 1=Block
||| 0=Permanent 1=Temporary
|| Clear Defect List
| 0=Append 1=Create

Figure 118
Command Packet for WRITE DEFECT LIST

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 8	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 119
Response Packet for WRITE DEFECT LIST

Table 100
Write Defect List Parameters 31-57

@	LTH	ID	OCTET	X/b	DEF	WRITE DEFECT LIST PARAMETERS	
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)	
			01-04			Count	
			05-08			Data Address	
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)	
			01-04			Residual Count	
			05-08			Data Address	
M	n+1	3E	01- n			PARTITION PARAMETER (See 5.5.15)	
M	n+1	55				DEFECTIVE DATABLOCK PARAMETER	
			01-02			Size of DataBlock Address Fields	
			03-			Address of Defective DataBlock	(first)
			n- :n			Address of Defective DataBlock	(last)
M	n+1	56				TRACK DEFECTS LIST PARAMETER	
			01-04			Cylinder	(first)
			05-06			Track	
			07-0A			Octet Offset into track	
			0B-0C			Length of defect (in bits)	
			n-B:8			Cylinder	(last)
			n-7:6			Track	
			n-5:2			Octet Offset into track	
			n-1:n			Length of defect (in bits)	
M	n+1	57				SECTOR DEFECTS LIST PARAMETER	
			01-04			Cylinder	(first)
			05-06			Track	
			07-08			No of Sector after Index	
			09-0C			Octet Offset within Sector	
			0D-0E			Length of defect (in bits)	
			n-D:A			Cylinder	(last)
			n-9:8			Track	
			n-7:6			No of Sector after Index	
			n-5:2			Octet Offset within Sector	
			n-1:n			Length of defect (in bits)	

11.4.4.1.6 Sector Defects List Parameter. See 8.7.6.3.2.

11.4.4.2 Parameters 58-6C. These parameters shall be as shown in Table 101.

11.4.4.2.1 Variable Size Sectors Parameter. See 11.3.4.2.1.

11.4.4.2.2 Cell Defects List Parameter. See 8.7.6.4.1.

11.4.4.2.3 Request Parm Parameter. See 6.3.4.13.1.

11.5 READ ERROR LOG

11.5.1 Command Packet. The command packet for this command shall be as shown in Figure 120.

11.5.2 Response Packet. The response packet for this command shall be as shown in Figure 121.

11.5.3 Description. The READ ERROR LOG command provides a method for the master to request statistics on usage information, error information, or both, from the addressee because it transfers the recorded error log data from the addressee. The error log contents and format are slave or facility specific.

Note that the error log contents may be returned as data containing the parameter list with the requested information. The error log data may be maintained automatically by the slave, maintained under direct control of the master, or by a combination of both.

If the addressee does not automatically clear error and usage counters after the successful completion of the command, the master shall specifically do so by using the Clear Error Log modifier. If the addressee does automatically clear the error log after successful completion of the command, the clearing of the log shall be inhibited when the Clear Error Log modifier is set.

Table 101
Write Defect List Parameters 58-6C

@	LTH	ID	OCTET	X/b	DEF	WRITE DEFECT LIST PARAMETERS
M	09	58				VARIABLE SIZE SECTORS PARAMETER
			01-02			Size of Cells at Index Gap
			03-04			Cell Size
			05-06			No of Cells Reserved per Track to Skip Flaws
			07-08			No of Cells to Skip per Flaw
M	n+1	5A				CELL DEFECTS LIST PARAMETER
			01-04			Cylinder (first)
			05-06			Track
			07-08			No of Defective Cells
			09-0C			Offset of Defective Cell #1
			0D- m			Offset of Defective Cell #2 through end
			n -			Cylinder (last)
			n -			Track
			n -			No of Defective Cells
			n -			Offset of Defective Cell #1
			n -			Offset of Defective Cell #2 through end
M	n+1	6C				REQUEST PARM PARAMETER
			01	7		Parameters as Data *
				6		Parameters in Response *
				5		Reserved
				4		Naked Parameters as Data *
				3-0		Reserved
			02			Parameter ID Repeated as many
			n			Parameter ID times as needed

* Mutually exclusive parameters.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	84	bbbb bbbb 7654 3210		xx	xx	

|||| Count 0=Octet 1=Block
||| Clear Error Log

Figure 120
Command Packet for READ ERROR LOG

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 121
Response Packet for READ ERROR LOG

Table 102
Read Error Log Parameters

@	LTH	ID	OCTET	X/b	DEF	READ ERROR LOG PARAMETERS
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)
			01-04			Count
			05-08			Data Address
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)
			01-04			Residual Count
			05-08			Data Address
S	n+1	50	01- n			ERROR LOG PARAMETER (vendor specific)
M	n+1	6C				REQUEST PARM PARAMETER
			01	7		Parameters as Data *
				6		Parameters in Response *
				5		Length *
				4		Naked Parameters as Data *
				3-0		Reserved
			02			Parameter ID Repeated as many
			n			Parameter ID times as needed
S	05	6D				PARM LENGTH PARAMETER
			01-04			Length of Parameter List

* Mutually exclusive parameters.

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD 4	SLAV ADDR 5	FAC ADDR 6	COMMAND PARAMETERS 6 through n
xxxx	xxxx	85	bbbb bbbb 7654 3210	xx		xx	

7654 3210
|||| Count 0=Octet 1=Block
|||
||
| 0=Append 1=Create

Figure 122
Command Packet for WRITE ERROR LOG

11.5.4 Parameters. The Read Error Log Parameters shall be as shown in Table 102.

11.5.4.1 Command Extent (Common) Parameter. See 5.5.2.

11.5.4.2 Response Extent (Common) Parameter. See 5.5.3.

11.5.4.3 Error Log Parameter. The field or fields in this parameter shall be defined in the vendor documentation.

11.5.4.4 Request Parm Parameter. See 6.3.4.13.1.

11.5.4.5 Parm Length Parameter. See 6.3.4.13.2.

11.6 WRITE ERROR LOG

11.6.1 Command Packet. The command packet for this command shall be as shown in Figure 122.

11.6.2 Response Packet. The response packet for this command shall be as shown in Figure 123.

11.6.3 Description. The WRITE ERROR LOG command transfers error log data to the addressee. The error log contents and format are slave or facility specific and may be transferred as data although it consists of the parameter list containing the required information to be logged.

The Create modifier clears any existing information in the addressee's error log, and resets any usage or error counters to zero.

11.6.4 Parameters. The Write Error Log Parameters shall be as shown in Table 103.

11.6.4.1 Command Extent (Common) Parameter. See 5.5.2.

11.6.4.2 Response Extent (Common) Parameter. See 5.5.3.

11.6.4.3 Error Log Parameter. See 11.5.4.1.3.

11.6.4.4 Request Parm Parameter. See 6.3.4.13.1.

11.7 DIAGNOSTIC CONTROL

11.7.1 Command Packet. The command packet for this command shall be as shown in Figure 124.

11.7.2 Response Packet. The response packet for this command shall be as shown in Figure 125.

Table 103
Write Error Log Parameters

@	LTH	ID	OCTET	X/b	DEF	WRITE ERROR LOG PARAMETERS	
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)	
			01-04			Count	
			05-08			Data Address	
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)	
			01-04			Residual Count	
			05-08			Data Address	
M	n+1	50	01- n			ERROR LOG PARAMETER (vendor specific)	
M	n+1	6C				REQUEST PARM PARAMETER	
			01	7		Parameters as Data	*
				6		Parameters in Response	*
				5		Reserved	
				4		Naked Parameters as Data	*
				3-0		Reserved	
			02			Parameter ID	Repeated as many
			n			Parameter ID	times as needed

* Mutually exclusive parameters.

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 8	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 123
Response Packet for WRITE ERROR LOG

PKT LTH	REF NO 0 1	OP CODE 2	COM MOD 3	OP MOD	SLAV ADDR 4	FAC ADDR 5	COMMAND PARAMETERS 6 through n
xxxx	xxxx	90	bbbb bbbb 7654 3210		xx	xx	

|||| Count 0=Octet 1=Block

Figure 124
Command Packet for DIAGNOSTIC CONTROL

PKT LTH	Echoed From Command 0 1 2 3 4 5	MAJOR STATUS CODES 6	TYPE 7	CODE 8	RESPONSE PARAMETERS 8 through n
xxxx	eeeeeeeeeeee	bbbb bbbb 7654 3210	0001	bbbb 3210	

Figure 125
Response Packet for DIAGNOSTIC CONTROL

Table 104
Diagnostic Control Parameters

@	LTH	ID	OCTET	X/b	DEF	DIAGNOSTIC CONTROL PARAMETERS	
M	09	31				COMMAND EXTENT PARAMETER (See 5.5.2)	
			01-04			Count	
			05-08			Data Address	
S	n+1	32				RESPONSE EXTENT PARAMETER (See 5.5.3)	
			01-04			Residual Count	
			05-08			Data Address	
M	03	50	01-02			DIAGNOSTIC NUMBER PARAMETER (vendor specific)	
M	n+1	51	01- n			DIAGNOSTIC FUNCTION LIST PARAMETER (vendor specific)	
M	n+1	52	01-02			DIAGNOSTIC ACTION CODE PARAMETER (vendor specific)	
			03- n			DIAGNOSTIC PARAMETERS (vendor specific)	
M	n+1	53	01- n			VENDOR-UNIQUE PARAMETER	
M	n+1	6C				REQUEST PARM PARAMETER	
			01	7		Parameters as Data	*
				6		Parameters in Response	*
				5		Reserved	
				4		Naked Parameters as Data	*
				3-0		Reserved	
			02			Parameter ID	Repeated as many
			n			Parameter ID	times as needed

* Mutually exclusive parameters.

11.7.3 Description. If there are no parameters, the DIAGNOSTIC CONTROL command causes the slave to transfer any diagnostic results currently in the slave as data.

The DIAGNOSTIC CONTROL command allows the master to load either a slave or a facility diagnostic. The diagnostic is transferred to the slave as data. Following loading, the selected slave or facility shall execute the supplied diagnostic. The slave shall execute according to the list. The parameters may result in transfers to or from the master, and, depending on the diagnostic sequences, both may occur during execution of this single command.

Results from the loaded diagnostic may be returned to the master as data. The parameters contained in the command packet and the results returned shall be defined in each product's functional specification.

This command may be used to read and write ID fields, to read and write data fields of sectors whose ID fields were damaged, and for other facility-type specific operations.

11.7.4 Parameters. The Diagnostic Control parameters shall be as shown in Table 104.

11.7.4.1 Command Extent (Common) Parameter. See 5.5.2.

11.7.4.2 Response Extent (Common) Parameter. See 5.5.3.

11.7.4.3 Diagnostic Number Parameter. See 11.1.4.3.

11.7.4.4 Diagnostic Function List Parameter. See 11.1.4.4.

11.7.4.5 Diagnostic Action Code Parameter. See 11.1.4.5.

11.7.4.6 Vendor-Unique Parameter. See 11.1.4.6.

11.7.4.7 Request Parm Parameter. See 6.3.4.13.1.

12. COMMAND SUMMARY

12.1 Control Commands

OP	COMMAND	OPCODE MODIFIERS
00	NOP	None
01	FACILITY OPERATION	None
02	ATTRIBUTES	x'0' - Report x'1' - Initialize x'2' - Restore x'9' - Load x'A' - Save
03	REPORT ADDRESSEE STATUS	0 - Condition 1 - Status 2 - Port Query
04	PORT ADDRESS	0 - Reserve/Release 1 - Priority Reserve 2 - Notify Alternate Ports of Priority Reserve
05	PATH CONTROL	0 - Purge Commands Outstanding at Excluded Port 1 - Path Select
06	ATTENTION CONTROL	x'0' - Enable x'2' - Disable x'4' - Clear x'6' - Set
07	OPERATING MODE	2 - Report/Set
08	ABORT	0 - Orderly Termination 1 - Terminate Command In Progress 2 - Terminate All Commands Not In Progress
09	ACCESS PERMITS	x'0' - Report x'1' - Initialize x'2' - Restore x'9' - Load x'A' - Save 2 - DataBlock/PhysicalBlock
0A	RESUME	None
0B	PORT RESPONSE	None
0C	ANTICIPATED ACTION	None

12.2 Position Commands

OP	COMMAND	OPCODE MODIFIERS
41	POSITION CONTROL	0 - Octet/Block Count 2 - DataBlock/PhysicalBlock
42	REPORT POSITION	0 - Octet/Block Count 2 - DataBlock/PhysicalBlock
44	REPORT DISCONTINUITY	0 - Octet/Block Count 1 - Search/List 2 - DataBlock/PhysicalBlock

12.3 Transfer Commands

OP	COMMAND	OPCODE MODIFIERS
10	READ	0 - Octet/Block Count 1 - Data Recovery On/Off 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
11	READ RAW DATA	0 - Octet/Block Count 1 - Data Recovery=1 (Off) 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
12	READ REPLICATED DATA	0 - Count=1 (Block) 1 - Data Recovery On/Off 2 - DataBlock/PhysicalBlock
18	SEARCH	0 - Octet/Block Count 1 - Data Recovery On/Off 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
20	WRITE	0 - Octet/Block Count 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
21	WRITE PATTERN	0 - Octet/Block Count 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
28	FORMAT	0 - Octet/Block Count 1 - Inhibit Defect Reallocation 2 - DataBlock/PhysicalBlock 3 - Initialize Format

12.4 Combination Commands

OP	COMMAND	OPCODE MODIFIERS
30	COPY	None
31	COMPARE SLAVE DATA	None
32	COMPARE DATA	None
33	REALLOCATE	0 - Relocate Data
34	ALLOCATE RESTORE	0 - Relocate Data
35	SHADOW READ	None
36	SHADOW WRITE	None
37	SHADOW RESTORE	None

12.5 Other Transfer Commands

OP	COMMAND	OPCODE MODIFIERS
50	READ VERIFY	0 - Octet/Block Count 1 - High Margin 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
51	READ AT FIRST AVAILABLE DATA	0 - Octet/Block Count 1 - Data Recovery On/Off 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
52	READ FROM BUFFER	None
53	READ FACILITY DATA TO BUFFER	0 - Octet/Block Count 1 - Data Recovery On/Off 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
54	READ PHYSICAL DATA AND ECC	0 - Octet/Block Count 1 - ECC/Syndrome 3 - Forward/Reverse
55	READ PHYSICAL HEADER	1 - Data Recovery On/Off 3 - Forward/Reverse

56	READ IPL	None
58	READ PHYSICAL HEADER & ECC	1 - ECC/Syndrome 3 - Forward/Reverse
62	WRITE TO BUFFER	None
63	WRITE BUFFER TO FACILITY	0 - Octet/Block Count 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
64	WRITE PHYSICAL DATA AND ECC	0 - Octet/Block Count 3 - Forward/Reverse
65	WRITE PHYSICAL HEADER	0 - Octet/Block Count 3 - Forward/Reverse
66	LOAD SLAVE IML	x'0' - Load x'1' - Reload x'2' - Report x'4' - Store
67	ERASE	0 - Octet/Block Count 2 - DataBlock/PhysicalBlock 3 - Forward/Reverse
68	WRITE PHYSICAL HEADER & ECC	0 - Octet/Block Count 3 - Forward/Reverse

12.6 Diagnostic Commands

OP	COMMAND	OPCODE MODIFIERS
80	PERFORM SLAVE DIAG	0 - Octet/Block Count
81	PERFORM FACILITY DIAG	0 - Octet/Block Count
82	READ DEFECT LIST	0 - Octet/Block Count 1 - Permanent/Temporary 2 - Working/Suspect 3 - Sequential/Chronological
83	WRITE DEFECT LIST	0 - Octet/Block Count 1 - Permanent/Temporary 2 - Clear Defect List 3 - Append/Create
84	READ ERROR LOG	0 - Octet/Block Count 1 - Clear Error Log

85	WRITE ERROR LOG	0 - Octet/Block Count 3 - Append/Create
90	DIAGNOSTIC CONTROL	0 - Octet/Block Count

X3.115-1984 Unformatted 80 Megabyte Trident Pack for Use at 370 tpi and 6000 bpi (General, Physical, and Magnetic Characteristics)
X3.116-1986 Recorded Magnetic Tape Cartridge, 4-Track, Serial 0.250 Inch (6.30 mm) 6400 bpi (252 bpmm), Inverted Modified Frequency Modulation Encoded
X3.117-1984 Printable/Image Areas for Text and Facsimile Communication Equipment
X3.118-1984 Financial Services — Personal Identification Number — PIN Pad
X3.119-1984 Contact Start/Stop Storage Disk, 158361 Flux Transitions per Track, 8.268 Inch (210 mm) Outer Diameter and 3.937 inch (100 mm) Inner Diameter
X3.120-1984 Contact Start/Stop Storage Disk
X3.121-1984 Two-Sided, Unformatted, 8-Inch (200-mm), 48-tpi, Double-Density, Flexible Disk Cartridge for 13 262 ftpr Two-Headed Application
X3.122-1986 Computer Graphics Metafile for the Storage and Transfer of Picture Description Information
X3.124-1985 Graphical Kernel System (GKS) Functional Description
X3.124.1-1985 Graphical Kernel System (GKS) FORTRAN Binding
X3.125-1985 Two-Sided, Double-Density, Unformatted 5.25-inch (130-mm), 48-tpi (1,9-tpmm), Flexible Disk Cartridge for 7958 bpr Use
X3.126-1986 One- or Two-Sided Double-Density Unformatted 5.25-inch (130-mm), 96 Tracks per Inch, Flexible Disk Cartridge
X3.127-1987 Unrecorded Magnetic Tape Cartridge for Information Interchange
X3.128-1986 Contact Start-Stop Storage Disk — 83 000 Flux Transitions per Track, 130-mm (5.118-in) Outer Diameter and 40-mm (1.575-in) Inner Diameter
X3.129-1986 Intelligent Peripheral Interface, Physical Level
X3.130-1986 Intelligent Peripheral Interface, Logical Device Specific Command Sets for Magnetic Disk Drive
X3.131-1986 Small Computer Systems Interface
X3.132-1987 Intelligent Peripheral Interface — Logical Device Generic Command Set for Optical and Magnetic Disks
X3.133-1986 Database Language —NDL
X3.135-1986 Database Language — SQL
X3.136-1986 Serial Recorded Magnetic Tape Cartridge for Information Interchange, Four and Nine Track
X3.139-1987 Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC)
X3.140-1986 Open Systems Interconnection — Connection Oriented Transport Layer Protocol Specification
X3.141-1987 Data Communication Systems and Services — Measurement Methods for User-Oriented Performance Evaluation
X3.146-1987 Device Level Interface for Streaming Cartridge and Cassette Tape Drives
X3.147-1987 Intelligent Peripheral Interface — Logical Device Generic Command Set for Magnetic Tapes
X3.153-1987 Open Systems Interconnection — Basic Connection Oriented Session Protocol Specification
X3.156-1987 Nominal 8-Inch Rigid Disk Removable Cartridge
X3.157-1987 Recorded Magnetic Tape for Information Interchange, 3200 CPI
X3.158-1987 Serial Recorded Magnetic Tape Cassette for Information Interchange, 0.150 Inch (3.81 mm), 8000 bpi (315 bpmm), Group Code Recording.
X11.1-1977 Programming Language MUMPS
IEEE 416-1978 Abbreviated Test Language for All Systems (ATLAS)
IEEE 716-1982 Standard C/ATLAS Language
IEEE 717-1982 Standard C/ATLAS Syntax
IEEE 770X3.97-1983 Programming Language PASCAL
IEEE 771-1980 Guide to the Use of ATLAS
ISO 8211-1986 Specifications for a Data Descriptive File for Information Interchange
MIL-STD-1815A-1983 Reference Manual for the Ada Programming Language
NBS-ICST 1-1986 Fingerprint Identification — Data Format for Information Interchange

---

X3/TR1-82 Dictionary for Information Processing Systems (Technical Report)

American National Standards for Information Processing

X3.1-1987 Synchronous Signaling Rates for Data Transmission
X3.2-1970 Print Specifications for Magnetic Ink Character Recognition
X3.4-1986 Coded Character Sets — 7-Bit ASCII
X3.5-1970 Flowchart Symbols and Their Usage
X3.6-1965 Perforated Tape Code
X3.9-1978 Programming Language FORTRAN
X3.11-1969 General Purpose Paper Cards
X3.14-1983 Recorded Magnetic Tape (200 CPI, NRZI)
X3.15-1976 Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission
X3.16-1976 Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.17-1981 Character Set for Optical Character Recognition (OCR-A)
X3.18-1974 One-Inch Perforated Paper Tape
X3.19-1974 Eleven-Sixteenths-Inch Perforated Paper Tape
X3.20-1967 Take-Up Reels for One-Inch Perforated Tape
X3.21-1967 Rectangular Holes in Twelve-Row Punched Cards
X3.22-1983 Recorded Magnetic Tape (800 CPI, NRZI)
X3.23-1985 Programming Language COBOL
X3.25-1976 Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.26-1980 Hollerith Punched Card Code
X3.27-1987 Magnetic Tape Labels and File Structure
X3.28-1976 Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links
X3.29-1971 Specifications for Properties of Unpunched Oiled Paper Perforator Tape
X3.30-1986 Representation for Calendar Date and Ordinal Date
X3.31-1973 Structure for the Identification of the Counties of the United States
X3.32-1973 Graphic Representation of the Control Characters of American National Standard Code for Information Interchange
X3.34-1972 Interchange Rolls of Perforated Tape
X3.37-1987 Programming Language APT
X3.38-1972 Identification of States of the United States (Including the District of Columbia)
X3.39-1986 Recorded Magnetic Tape (1600 CPI, PE)
X3.40-1983 Unrecorded Magnetic Tape (9-Track 800 CPI, NRZI; 1600 CPI, PE; and 6250 CPI, GCR)
X3.41-1974 Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange
X3.42-1975 Representation of Numeric Values in Character Strings
X3.43-1986 Representations of Local Time of Day
X3.44-1974 Determination of the Performance of Data Communication Systems
X3.45-1982 Character Set for Handprinting
X3.46-1974 Unrecorded Magnetic Six-Disk Pack (General, Physical, and Magnetic Characteristics)
X3.47-1977 Structure for the Identification of Named Populated Places and Related Entities of the States of the United States for Information Interchange
X3.48-1986 Magnetic Tape Cassettes (3.81-mm [0.150-Inch] Tape at 32 bpmm [800 bpi], PE)
X3.49-1975 Character Set for Optical Character Recognition (OCR-B)
X3.50-1986 Representations for U.S. Customary, SI, and Other Units to Be Used in Systems with Limited Character Sets
X3.51-1986 Representations of Universal Time, Local Time Differentials, and United States Time Zone References
X3.52-1976 Unrecorded Single-Disk Cartridge (Front Loading, 2200 BPI) (General, Physical, and Magnetic Requirements)
X3.53-1976 Programming Language PL/I
X3.54-1986 Recorded Magnetic Tape (6250 CPI, Group Coded Recording)
X3.55-1982 Unrecorded Magnetic Tape Cartridge, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase encoded
X3.56-1986 Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.57-1977 Structure for Formatting Message Headings Using the American National Standard Code for Information Interchange for Data Communication Systems Control
X3.58-1977 Unrecorded Eleven-Disk Pack (General, Physical, and Magnetic Requirements)
X3.60-1978 Programming Language Minimal BASIC
X3.61-1986 Representation of Geographic Point Locations
X3.62-1987 Paper Used in Optical Character Recognition (OCR) Systems
X3.63-1981 Unrecorded Twelve-Disk Pack (100 Megabytes) (General, Physical, and Magnetic Requirements)
X3.64-1979 Additional Controls for Use with American National Standard Code for Information Interchange
X3.66-1979 Advanced Data Communication Control Procedures (ADCCP)
X3.72-1981 Parallel Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.73-1980 Single-Sided Unformatted Flexible Disk Cartridge (for 6631-BPR Use)
X3.74-1987 Programming Language PL/I, General-Purpose Subset
X3.76-1981 Unformatted Single-Disk Cartridge (Top Loading 200 tpi 4400 bpi) (General, Physical, and Magnetic Requirements)
X3.77-1980 Representation of Pocket Select Characters
X3.78-1981 Representation of Vertical Carriage Positioning Characters in Information Interchange
X3.79-1981 Determination of Performance of Data Communications Systems That Use Bit-Oriented Communication Procedures
X3.80-1981 Interfaces between Flexible Disk Cartridge Drives and Their Host Controllers
X3.82-1980 One-Sided Single-Density Unformatted 5.25-Inch Flexible Disk Cartridge (for 3979-BPR Use)
X3.83-1980 ANSI Sponsorship Procedures for ISO Registration According to ISO 2375
X3.84-1981 Unformatted Twelve-Disk Pack (200 Megabytes)(General, Physical, and Magnetic Requirements
X3.85-1981 1/2-Inch Magnetic Tape Interchange Using a Self Loading Cartridge
X3.86-1980 Optical Character Recognition (OCR) Inks
X3.88-1981 Computer Program Abstracts
X3.89-1981 Unrecorded Single-Disk, Double-Density Cartridge (Front Loading, 2200 bpi, 200 tpi) (General, Physical, and Magnetic Requirements)
X3.91M-1987 Storage Module Interfaces
X3.92-1981 Data Encryption Algorithm
X3.93M-1981 OCR Character Positioning
X3.94-1985 Programming Language PANCM
X3.95-1982 Microprocessors — Hexadecimal Input/Output, Using 5-Bit and 7-Bit Teleprinters
X3.96-1983 Continuous Business Forms (Single-Part)
X3.98-1983 Text Information Interchange in Page Image Format (PIF)
X3.99-1983 Print Quality Guideline for Optical Character Recognition (OCR)
X3.100-1983 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Packet Mode Operation with Packet Switched Data Communications Network
X3.101-1984 Interfaces Between Rigid Disk Drive(s) and Host(s)
X3.102-1983 Data Communication Systems and Services — User-Oriented Performance Parameters
X3.103-1983 Unrecorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in)
X3.104-1983 Recorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in), Phase Encoded
X3.105-1983 Data Link Encryption
X3.106-1983 Modes of Operation for the Data Encryption Algorithm
X3.110-1983 Videotex/Teletext Presentation Level Protocol Syntax
X3.111-1986 Optical Character Recognition (OCR) Matrix Character Sets for OCR-M
X3.112-1984 14-in (356-mm) Diameter Low-Surface-Friction Magnetic Storage Disk
X3.113-1987 Programming Language FULL BASIC
X3.114-1984 Alphanumeric Machines; Coded Character Sets for Keyboard Arrangements in ANSI X4.23-1982 and X4.22-1983

(Continued on reverse)

October 1987

NIST-772
(REV. 10-88)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

CHANGE NUMBER 2 - FIPS 60-2 & 62
1-FIPS 61-1,63-1,97,111,130&131

DATE OF CHANGE
1990 December 26

FIPS PUBLICATION NUMBER
See above.

FIPS PUBLICATION CHANGE NOTICE

PUBLICATION TITLE FIPS 60-2, I/O Channel Interface; 62, Operational Specifications for Magnetic Tape Subsystems; 61-1, Channel Level Power Control Interface; 63-1, Operational Specifications for Variable Block Rotating Mass Storage Subsystems; 97, Operational Specifications for Fixed Block Rotating Mass Storage Subsystems; 111, Storage Module Interfaces (w/extens. for enhanced storage module interface); 130, Intelligent Peripheral Interface (IPI); 131, Small Computer System Interface (SCSI).

THIS OFFICE HAS A RECORD OF YOUR INTEREST IN RECEIVING CHANGES TO THE ABOVE FIPS PUBLICATION. THE CHANGE(S) INDICATED BELOW HAVE BEEN PROVIDED BY THE MAINTENANCE AGENCY FOR THIS PUBLICATION AND WILL BE INCLUDED IN THE NEXT PUBLISHED REVISION TO THIS FIPS PUBLICATION. QUESTIONS OR REQUESTS FOR ADDITIONAL INFORMATION SHOULD BE ADDRESSED TO THE MAINTENANCE AGENCY:

Department of Commerce
National Institute of Standards and Technology
National Computer Systems Laboratory
Gaithersburg, MD 20899

CHANGE ITEM(S)

Attached is a reprint from the December 18, 1990, FEDERAL REGISTER (55 FR 51941) which provides approved revisions by the Secretary of Commerce to the FIPS family of input/output interface standards, and the approved discontinuation of the Exclusion and Verification Lists for these standards.

These approved revisions became effective on December 18, 1990, and become an integral part of FIPS 60-2, 61-1, 62, 63-1, 97, 111, 130 and 131, and, as such, are considered to be included whenever reference is made to them.

These approved revisions should be filed with each FIPS listed above.

Attachment

Copies of FIPS are available from:

National Technical Information Service (NTIS)
ATTN: Sales Office, Sills Building
5285 Port Royal Road
Springfield, Virginia 22161

Phone - 703/487-4650 Office Hours - 7:45 a.m. to 4:15 p.m.

ELECTRONIC FORM

12-18-90
Vol. 55 No. 243
Pages 51895-52036

Federal Register

Tuesday
December 18, 1990

National Institute of Standards and Technology

[Docket No. 900101-0219]

RIN 0693-AA59

Approval of Revisions to Federal Information Processing Standards (FIPS) Family of Input/Output Interface Standards

AGENCY: National Institute of Standards and Technology (NIST), Commerce.

ACTION: The purpose of this notice is to announce that the Secretary of Commerce has approved revisions to the Federal Information Processing Standards (FIPS) family of input/output interface standards, and has approved discontinuation of the exclusion and verification lists for these standards.

SUMMARY: On March 20, 1990, notice was published in the Federal Register (55 FR 10272) proposing revision of Federal Information Processing Standards (FIPS) 60-2, 61-1, 62, 63-1, 97, 111, 130, and 131 to make them non-mandatory, and discontinue the exclusion and verification lists for these standards. This proposal superseded the proposal for revision of these standards announced in the Federal Register (52 FR 44462) of November 19, 1987. Procedures for the Exclusion List for FIPS 60, 61, 62, 63, and 97 were published in the Federal Register on

September 3, 1982 (47 FR 38959–38960). Procedures for the Verification List for FIPS 60, 61, 62, 63, and 97 were published in the Federal Register on December 11, 1979 (44 FR 71444–71445) and on April 7, 1981 (46 FR 20719–20720).

The written comments submitted by interested parties and other material available to the Department relevant to these proposed revisions were reviewed by NIST. On the basis of this review, NIST recommended that the Secretary approve revisions to the input/output family of standards and approve discontinuation of the exclusion and verification lists for these standards. NIST prepared a detailed justification document for the Secretary's review in support of those recommendations.

This notice provides only the changes to the revised standards.

EFFECTIVE DATE: These revisions are effective December 18, 1990.

ADDRESSES: Interested parties may obtain copies of FIPS PUBS 60–2, 61–1, 62, 63–1, 97, 111, 130, and 131 from the National Technical Information Service, U.S. Department of Commerce, Springfield, VA 22161.

FOR FURTHER INFORMATION CONTACT: Ms. Shirley Radack, National Institute of Standards and Technology, Gaithersburg, MD 20899, telephone (301) 975–2833.

SUPPLEMENTARY INFORMATION: Under the provisions of 40 U.S.C. 759(d), the Secretary of Commerce is authorized to promulgate standards and guidelines for Federal computer systems, and to make such standards compulsory and binding to the extent to which the Secretary determines necessary to improve the efficiency of operation, or security and privacy of Federal computer systems.

The family of I/O interface standards currently includes:

a. FIPS 60–2, I/O Channel Interface, revised July 29, 1983.

b. FIPS 61–1, Channel Level Power Control Interface, revised July 13, 1982.

c. FIPS 62, Operational Specifications for Magnetic Tape Subsystems, revised December 30, 1980.

d. FIPS 63–1, Operational Specifications for Variable Block Rotating Mass Storage Subsystems, revised April 14, 1983; Supplement to FIPS PUB. 63–1, Additional Operational Specifications for Variable Block Rotating Mass Storage Subsystems, April 14, 1983.

e. FIPS 97, Operational Specifications for Fixed Block Rotating Mass Storage Subsystems, February 4, 1983.

f. FIPS 111, Storage Module Interfaces (with extensions for enhanced storage module interfaces), April 18, 1985.

g. FIPS 130, Intelligent Peripheral Interface (IPI), July 16, 1987.

h. FIPS 131, Small Computer System Interface (SCSI) July 16, 1987.

The following revisions are being made effective immediately upon publication. A delayed effective date is not required because these standards are exempt from the Administrative Procedure Act by U.S.C. 553(a)(2).

Revisions to Federal Information Processing Standards 60–2, 61–1, 62, 63–1, 97, 111, 130, and 131.

FIPS 60–2, I/O Channel Interface, is revised as follows:

Applicability. This standard addresses the interconnection of computer peripheral equipment as a part of ADP systems for the following types of peripherals: (1) Magnetic tape equipment employing open reel-to-reel magnetic tape storage devices, specifically excluding magnetic tape cassette and tape cartridge storage devices, (2) magnetic disk storage equipment employing disk drives each having a capacity greater than 7 megabytes per storage module, excluding flexible disk and disk cartridge devices having a smaller storage capacity per device, and (3) other peripheral equipment employing peripheral device types for which operational specifications standards have been issued as Federal Information Processing Standards. This standard is recommended for use in the acquisition of peripheral equipment for ADP systems with input/output channel interfaces as specified in the technical specifications, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective December 13, 1979. The first revision became effective June 23, 1980, and the second revision became effective July 29, 1983. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 61–1, Channel Level Power Control Interface, is revised as follows:

Applicability. This standard addresses the power control interface in connecting computer peripheral equipment to ADP systems. It is recommended for use when FIPS 60–2 is used, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective June 23, 1980, and the first revision became effective July 13, 1982. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 62, Operational Specifications for Magnetic Tape Subsystems, is revised as follows:

Applicability. This standard addresses magnetic tape equipment connected to ADP systems through FIPS 60 interfaces. It is recommended for use in the acquisition of such equipment, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective June 23, 1980. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 63–1, Operational Specifications for Variable Block Rotating Mass Storage Subsystems, is revised as follows:

Applicability. This standard addresses peripheral device dependent operational interfaces for connecting variable block rotating mass storage equipment to ADP systems through FIPS 60 interfaces. It is recommended for use in the acquisition of such variable block rotating mass storage equipment for connection to ADP systems, when it is determined that interchange of equipment between different systems is likely.

Implementation. This standard became effective June 23, 1980, and the first revision became effective April 14, 1983. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 97, Operational Specifications for Fixed Block Rotating Mass Storage Subsystems, is revised as follows:

Applicability. This standard addresses the peripheral device dependent operational interface specifications for connecting fixed block rotating mass storage equipment to ADP systems through FIPS 60 interfaces. It is recommended for use in the acquisition of such fixed block rotating mass storage equipment for connection to ADP systems, when it is determined that interchange of equipment between different systems is likely.

Implementation. The original version of this standard became effective February 4, 1983. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 111, Storage Module Interfaces, is revised as follows:

Applicability. This standard addresses connection of a disk drive to a controller as part of an ADP system. This standard is recommended for use in the acquisition of disk systems that are

connected to small and medium sized computer systems, when it is determined that interchange of equipment between different systems is likely.

Implementation. This standard became effective May 18, 1985. This revision becomes effective December 18, 1990.

Waivers. This standard is non-mandatory. No waivers are required.

FIPS 130, Intelligent Peripheral Interface (IPI), is revised as follows:

Section 8, Applicability. This standard applies to the connection of computers to storage peripheral device controllers. This standard is recommended for use in the acquisition of magnetic disk drives, optical disk drives, and tape drives to be connected to minicomputer systems, when it is determined that interchange of equipment between different systems is likely.

Section 10, Implementation. This standard became effective December 16, 1987. This revision becomes effective December 18, 1990.

Section 11, Waivers. This standard is non-mandatory. No waivers are required.

FIPS 131, Small Computer System Interface (SCSI) is revised as follows:

Section 8, Applicability. This standard addresses the connection of small computers to peripheral devices with integral controllers. This standard is recommended for use in the acquisition of storage peripherals and small computer systems for office or laboratory use, when it is determined that interchange of equipment between different systems is likely.

Section 10, Implementation. This standard became effective December 16, 1987. This revision becomes effective December 18, 1990.

Section 11, Waivers. This standard is non-mandatory. No waivers are required.

Dated: December 12, 1990.

John W. Lyons,

Director.

[FR Doc. 90-29563 Filed 12-17-90; 8:45 am]

BILLING CODE 3510-CN-M