DOCUMENT STATUS

REVISION:	1.0
DATE:	5/19/1997
EDITOR:	Michael Poulsen, Darrell J. Redford
COMPANY/OWNER	Iomega Corporation

REMOVABLE / RE-WRITABLE MEDIA SPECIFICATION

Draft Proposal

Point of Contact

Michael Poulsen – Engineer
Darrell Redford - Technical Evangelist
Iomega Corporation
1821 West Iomega Way, Roy UT 84067
Ph: (801) 778-4245 778-4432
Fax: (801) 778-4285
E-Mail: poulsenm@iomega.com
redfordd@imoega.com

CONTENTS

1.1 Introduction

This document defines a standard method for providing Removable/Re-Writable Media support for devices utilizing either

A) the existing ATA host computer hardware and cabling and ATAPI protocol or
B) B) packetized command protocol utilizing command wrappers (ATAPI Packet, USB, 1394, etc.).

1.2 Scope

This document is intended to be used in conjunction with the ATAPI Removable Rewritable Media document SFF-8070, Commands for DVD devices document SFF-8090 and Small Computer System Interface (SCSI-2) specification document X3T9.2 or later.
In addition it proposes areas of enhanced functionality to these documents in order to support Removable/Re-Writable Media.

1.3 Related Documents

ATAPI Removable Rewritable Media SFF-8070
Commands for DVD Devices SFF-8090
SCSI-2 Specification X3T9.2 or later.

1.4 Document Conventions

The terms 'Host Computer', 'Initiator' are synonymous and refer to the controller platform from which commands are constructed and initiated. The Host Computer or Initiator sends commands to the Target Device.

The terms 'Target Device', 'Target', 'Device' are synonymous and refer to the unit that controls and accesses the Removable/Re-Writable Media. The Target Device receives commands from the Host Computer, executes those commands and returns statuses to the Host Computer, indicating the results of the commands.

1.5 Abbreviations

TBD – To Be Determined

1.6 Unresolved Issues

Return and error reporting conventions and statues for each individual command.

2.0 Command Set

2.1 FORMAT UNIT Command

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

Table 2.1.1 FORMAT UNIT COMMAND

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (04h)							
1	Logical Unit Number			FmtData	CmpLst	Defect List Format		
2	Vendor Specific							
3-4	Interleave							
5	Control							
6-11	Pad (0)							

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

During the format operation, the target ***shall*** respond to the following commands as follows:

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

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

c) In response to the REQUEST SENSE command, unless an error has occurred, the target shall return a sense key of NOT READY and an additional sense code of LOGICAL UNIT NOT READY FORMAT IN PROGRESS, with the sense key specific bytes set for progress indication.

During the execution of the FORMAT UNIT command, the target may perform a medium defect management algorithm. Four sources of defect location information are defined as follows:

- **PLIST** Primary Defect List
- **CLIST** List of defects detected by the target during the formatting process
- **DLIST** Defect Descriptor List
- **GLIST** Grown Defect List

FmtData bit:

1 = DATA OUT phase will occur. The initiator will send a Defect List header followed by a Defect List of specified length.

0 = No DATA OUT phase will occur and no Defect List will be sent.

CmpLst bit:

1 = The Defect List sent is a complete list of defective blocks. All blocks that were previously field flagged, reassigned, or reallocated will be reformatted; these blocks may be returned to use.

0 = All Defect List blocks (along with all field flagged, reassigned, and reallocated blocks) will be unconditionally flagged and replaced with spares.

Defect List Format:

Refer to SCSI Spec X3T9.2 or later.

Interleave

Refer to SCSI Spec X3T9.2 or later.

2.2 Get Event Status Notification

2.3 Inquiry

The INQUIRY command requests that information regarding parameters of the target and its attached peripheral devices be sent to the initiator.

Table 2.3.1 INQUIRY COMMAND

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (12h)							
1	Logical Unit Number			Reserved				
2-3	Reserved							
4	Allocation Length							
6-11	Pad (0)							

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

If an INQUIRY command is received from an initiator with a pending unit attention condition, the target shall perform the INQUIRY command and shall not clear the unit attention condition.

Allocation Length:

The number of bytes of information requested from the logical device up to a maximum of n bytes (for n specified in Table 2.3.2 below).

Depending upon device specific implementation, the logical device may return fewer than the requested n bytes. This is not considered an error.
An Allocation Length of zero (0) should return no data.

The standard INQUIRY data contains 36 required bytes, followed by a variable number of vendor specific parameters.

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

Table 2.3.2 Standard INQUIRY data format

	Bit							
Byte	7	6	5	4	3	2	1	0
0	Peripheral Qualifier			Peripheral Device Type				
1	RMB	Reserved						
2	ISO Version		ECMA Version			ANSI Version		
3	AENC	Reserved		MESN	Response Data Format			
4	Additional Length							
5-6	Reserved							
7	RelAdr	Wbus32	Wbus16	Sync	Linked	Rsrvd	CmdQue	SftRe
8-15	Vendor Identification							
16-31	Product Identification							
32-35	Product Revision Level							
36-55	Vendor Specific							
56-95	Reserved							
Vendor Specific Parameters								
96-n	Vendor Specific							

The peripheral qualifier and peripheral device type fields identify the device currently connected to the logical unit. If the target is not capable of supporting a device on this logical unit, this field shall be set to 7Fh (peripheral qualifier set to 011b and peripheral device type set to 1Fh). The peripheral qualifier is defined below in table 2.3.3 and the peripheral device type in table 2.3.4.

Table 2.3.3 PERIPHERAL QUALIFIERS

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

Table 2.3.4 PERIPHERAL DEVICE TYPES

Code	Description
00h	Direct-Access device
01h-06h	Reserved
07h	Optical memory device
08-1Eh	Reserved
1Fh	Unknown or no device type

RMB bit:

A Removable Medium Bit (RMB) of one indicates that the medium is removable. A RMB bit of zero (0) indicates that the medium is not removable.

ISO bit and ECMA bits:

The use of non-zero code values in the ISO version and ECMA version fields are defined by the International Organization for Standardization and the European Computer Manufactures Association, respectively.

ANSI bit:

Refer to SCSI Spec X3T9.2 or later.

AENC bit:

Asynchronous Event Notification Capability. If this bit is set, the device supports Asynchronous Event Notification. If this bit is not set, the device does not support Asynchronous Event Notification.

NOTE: If the MESN bit is set and this bit contains a 0, then only polling Event Notification capability is supported.

MESN bit:

This bit is used in conjunction with the AENC bit.
This bit indicates that the logical device supports MESN (Media Event Status Notification). A value of one (1) in this bit indicates that MESN is supported for the logical device. A value of zero (0) indicates that MESN is not supported. If this bit is set, then the AENC bit takes on the following meanings:

AENC = 1 – MESN is supported in asynchronous mode.
AENC = 0 – MESN is supported in polling mode.

Response Data Format field:

A response data format value of one or two (1 or 2) indicates that the data format conforms to this specification. Any other data format values are reserved.

Additional Length field:

This field specifies the length in bytes of the parameters. It is equal to the total number of bytes contained within the INQUIRY data minus four (4).

If the allocation length of the Command Packet is too small to transfer all of the parameters, the additional length shall not be adjusted to reflect the truncation.

Vendor Identification field:

The Vendor Identification field contains 8 bytes of ASCII data identifying the vendor of the product. The data shall be left aligned within this field.

Product Identification field:

The Product Identification field contains 16 bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

Product Revision Level field:

The Product Revision Level field contains 4 bytes of ASCII data as defined by the vendor. The data shall be left-aligned within this field.

2.4 Mode Select

The Mode Select command provides a means for the initiator to specify medium, logical unit, or peripheral device parameters to the target.

NOTE: Initiators should issue MODE SENSE prior to each MODE SELECT to determine supported mode pages, mode page lengths, and other parameters. Targets that implement the MODE SELECT command shall also implement the MODE SENSE command.

Table 2.4.1 MODE SELECT COMMAND

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (55h)							
1	Logical Unit Number			PF (1)		Reserved		SP
2-6	Reserved							
7	Parameter List Length (MSB)							
8	Parameter List Length (LSB)							
9-11	Pad (0)							

See the Mode Sense command for descriptions of the Mode Page Parameters.

PF bit:

The PF bit should always be set to 1, indicating that the MODE SELECT parameters conform to the X3T9.2 SCSI standard.

SP bit:

A Save Pages bit of zero indicates the logical unit shall perform the specified MODE SELECT operation and shall not save any pages. A Save Pages bit of one indicates that the logical unit shall perform the specified MODE SELECT operation and shall save to a non-volatile, vendor-specific location all of the savable pages. If the logical unit supports saved pages, it shall save only one copy of the page. The SP bit is optional, even when mode pages are supported by the target. Pages that are saved are identified by the parameter savable bit that is returned in the page header by the MODE SENSE command. If the PS bit is set in the MODE SENSE data, then the page shall be savable by issuing the MODE SELECT command with the SP bit set. If the target does not implement saved pages and the SP bit is set to on, the command shall be terminated with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN COMMAND PACKET.

Parameter List Length fields:

The Parameter List Length fields specify the length in bytes of the mode parameter list that shall be transferred from the initiator to the target after the command packet has been transferred. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered an error.

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

2.5 Mode Sense

The MODE SENSE command provides a means for a target to report parameters to the initiator. It is a complementary command to the MODE SELECT command.

Targets that implement the MODE SENSE command shall also implement the MODE SELECT command.

Table 2.5.1 MODE SENSE COMMAND

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (5Ah)							
1	Logical Unit Number			Reserved	DBD	Reserved		
2	PC		Page Code					
3-6	Reserved							
7	Parameter List Length (MSB)							
8	Parameter List Length (LSB)							
9-11	Pad (0)							

DBD bit:

The DBD bit controls whether the drive sends the block descriptor.

DBD = 1 The drive does not send the block descriptor
DBD = 0 The drive sends the block descriptor

Refer to table 2.5.6 for more information on the Block Descriptor format.

PC bit:

The PC (Page Control) bit defines the type of mode parameter values to be returned within the mode pages.
Refer to table 2.5.2 below for a list of supported PC values.

Table 2.5.2 Page Control Field Values

Code	Type of Parameter	Reference Location
00b	Current Values	???
01b	Changeable Values	???
10b	Default Values	???
11b	Saved Values	???

Page Code field:

The Page Code specifies which mode page(s) to return. Mode pages shall be returned in ascending page code order except for mode page 00h.
Refer to table 2.5.3 below for information regarding supported Page Codes.

Table 2.5.3 Page Code Values

Page Code	Description	Reference Location
00h	Vendor Specific (does not require page format)	Refer to SCSI Spec
01h	Read-Write Error Recovery Page	Refer to SCSI Spec
02h	Disconnect-Reconnect Page	Refer to SCSI Spec
03h	Format Device Page	Refer to SCSI Spec
04h	Rigid Disk Drive Geometry Page	Refer to SCSI Spec
05h	Flexible Disk Page	Refer to SCSI Spec
06h	Reserved	NA
07h	Verify Error Recovery Page	Refer to SCSI Spec
08h	Caching Page	Refer to SCSI Spec
09h	Peripheral Device Page	Refer to SCSI Spec
0Ah	Control Mode Page	Refer to SCSI Spec
0Bh	Medium Types Supported Page	Refer to SCSI Spec
0Ch	Notch and Partition Page	Refer to SCSI Spec
0Dh-3Eh	Reserved	NA
3Fh	Return all pages (valid for MODE SENSE only)	NA

The Mode Parameter List is sent to the initiator for the MODE SENSE command and to the target for the MODE SELECT command. The mode parameter list consists of a mode parameter header, zero or more block descriptors, and zero or more mode pages.

Mode Parameter Headers contain 4 bytes each and are required for both the MODE SENSE and MODE SELECT commands.

Refer to tables 2.5.4 and 2.5.5 for information on the Mode Parameter Headers.

Table 2.5.4 MODE SENSE PARAMETER HEADER

	Bit							
Byte	7	6	5	4	3	2	1	0
0	Mode Data Length							
1	Media Type							
2	WP	Reserved			1	Reserved		
3	Block Descriptor Length							

Mode Data Length field:

The Mode Data Length field specifies the length in bytes of the data available for transfer to the initiator. Mode data length equals three plus the value in the Block Descriptor Length field plus the length of all requested pages.

Media Type field:

This field represents the media type in use (i.e. 0h for Direct-access, 07h for Optical).

WP bit:

The WP (Write Protect) bit reflects the write-protect status of the media currently loaded in the logical unit.

1 = Writes are rejected
0 = Writes are accepted

Block Descriptor Length field:

This field specifies the number of bytes in the Block Descriptor. Valid values are zero and eight.

Table 2.5.5 MODE SELECT PARAMETER HEADER

	Bit							
Byte	7	6	5	4	3	2	1	0
0	Reserved							
1	Media Type							
2	Reserved							
3	Block Descriptor Length							

Media Type field:

Refer Media Type field under MODE SENSE PARAMETER HEADER

Block Descriptor Length field:

Refer Media Type field under MODE SENSE PARAMETER HEADER

Table 2.5.6 MODE PARAMETER BLOCK DESCRIPTOR

	Bit							
Byte	7	6	5	4	3	2	1	0
0-4	Reserved							
5-8	(MSB) Block Length (LSB)							

Block Length field:

For MODE SELECT this value must be equal to the Block Length returned by the drive from the MODE SENSE command.

For MODE SENSE this value will contain the length in bytes of each block as formatted on the media.

NOTE: The descriptor block is sent only if the DBD bit in the MODE SENSE command is zero and the Block Descriptor Length is eight.

2.6 Prevent/Allow Medium Removal

The PREVENT/ALLOW MEDIUM REMOVAL command requests that the logical unit enable or disable the removal of the medium. The prevention of media removal (when implemented) shall be accomplished through the use of a locking mechanism. The use of a physical locking mechanism is optional. If a Non-Persistent Prevent is issued, and the logical unit does not support a physical locking mechanism, an error shall be returned (ILLEGAL REQUEST/INVALID FIELD IN COMMAND PACKET). If the operation is persistent, then the Prevent will not be reset when media is removed or inserted. This

will allow new media to become captive without host interaction. The Persistent Prevent is to be used in conjunction with the Get Event Status Notification command, to prevent media from being ejected with dirty file system buffers (or cache).

Table 2.6.1 PREVENT/ALLOW MEDIUM REMOVAL

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (1Eh)							
1	Logical Unit Number			Reserved				
2-3	Reserved							
4	Reserved						Persist	Prevent
5	Control							
6-11	Pad (0)							

Persist bit:

The Persist bit enables the persistent capabilities needed for MESN support. A Persist bit set to one (1) indicates that this will be a Persistent Prevent/Allow command. If the Prevent and Persistent bits are both one (1), upon receiving this command, the target shall disable any hardware eject mechanisms. Furthermore, all media after initial drive spin up, shall remain locked in the drive until the host issues an eject request, or Persistent Prevent status is reset and the hardware eject mechanism again becomes available.

The Persistent Prevent status shall be reset upon receipt of a Persistent Allow command, a Bus Reset, or a Power Reset condition.

Upon insertion of new media, under Persistent Prevent conditions, the target eject controls shall remain functional up until the drive generates a New Media event as defined in the Media Status Events section (Section 3.0). After this event has been generated, the media shall remain locked as defined above.

NOTE: The Prevent command with a Persist bit of one (1) shall not prevent an eject request from the host from succeeding.

The behavior of the Prevent/Allow command with a Persistent bit of zero (0) is not effected by the Persistent Prevent state. The prevention of medium removal shall begin when the Initiator issues a PREVENT/ALLOW MEDIUM REMOVAL command with a Prevent bit of one and a Persist bit of zero (medium removal prevented). The prevention of medium removal shall terminate under the following conditions:

1. After the initiator has issued a PREVENT ALLOW MEDIUM REMOVAL command with a Prevent bit of zero (unlock), and the logical unit has successfully performed a Flush cache operation.

2. Upon a hard RESET condition.

3. Upon a DEVICE RESET in an ATAPI environment.

4. If the device does not support a locking mechanism.

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

The default state of the drive at power on is unlocked, unless the drive supports a Prevent/Allow jumper and the jumper is in the Prevent state.

This command will affect the actions of the START/STOP UNIT command according to the following table, table 2.6.2:

NOTE: 0 = Not enabled (except for Start which means spin down).
1 = Enabled (except for Start which means spin up).

Table 2.6.2 PREVENT/EJECT/START COMBINATIONS

Prevent	LoEj	Start	Medium Locked	Medium Present	Action
0	0	0	0	0	No Action
0	0	0	0	1	No Action/Spin Down
0	0	0	1	0	Not Allowed
0	0	0	1	1	Spin Down
0	0	1	0	0	Error: Medium Not Present
0	0	1	0	1	Spin Up
0	0	1	1	0	Not Allowed
0	0	1	1	1	spin Up
0	1	0	0	0	No Action
0	1	0	0	1	Spin Down/Eject
0	1	0	1	0	Not Allowed
0	1	0	1	1	Error: Medium Removal Prevented
0	1	1	0	0	Error: Medium Not Present
0	1	1	0	1	Spin Up
0	1	1	1	0	Not Allowed
0	1	1	1	1	Spin Up
1	0	0	0	0	Error: Medium Not Present
1	0	0	0	1	Prevent
1	0	0	1	0	Not Allowed
1	0	0	1	1	Prevent
1	0	1	0	0	Error: Medium Not Present
1	0	1	0	1	Spin Up/Prevent
1	0	1	1	0	Not Allowed
1	0	1	1	1	Spin Up/Prevent
1	1	0	0	0	Error: Medium Not Present
1	1	0	0	1	See Item 1 below
1	1	0	1	0	Not Allowed
1	1	0	1	1	See Item 2 below
1	1	1	0	0	Error: Medium Not Present
1	1	1	0	1	Spin Up/Prevent
1	1	1	1	0	Not Allowed
1	1	1	1	1	Spin Up/Prevent

Item 1:

If Prevent is issued before eject, then the Prevent action should be performed normally. When the eject action is performed in the START/STOP UNIT command, an error should be issued, since the medium is under prevention status from the previous command.

If the eject action is performed before the Prevent action, then the medium should be ejected normally. When the Prevent action is then issued via the PREVENT/ALLOW MEDIUM REMOVAL command, an error should be issued since no medium will then be present.

Item 2:

If Prevent is issued before eject, then no action will be taken. When the eject action is then issued via the START/STOP UNIT command, an error should be issued since the medium will be under Prevent status.

If the eject action is issued before the Prevent action, an error will be generated since the medium will be under Prevent status.

2.7 Read Capacity

The READ CAPACITY command provides a means for the initiator to request information regarding the capacity of the installed media.

Table 2.7.1 READ CAPACITY

	Bit							
Byte	**7**	6	5	4	3	2	1	0
0	OPCODE (25h)							
1	Logical Unit Number			Reserved				
2-5	(MSB) Logical Block Address (LSB)							
6-7	Reserved							
8	Reserved							PMI
9	Control							
10-11	Pad (0)							

Logical Block Address field:

The logical block address field is used exclusively in conjunction with the PMI bit. Refer to the definition of the PMI bit below for a definition of the Logical Block Address field.

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

PMI bit:

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

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

The format for the returned data is shown in table 2.7.2 below.

Table 2.7.2 Partial Medium Indicator Bit (PMI)

	Bit							
Byte	7	6	5	4	3	2	1	0
0-3	(MSB)	Returned Logical Block Address						(LSB)
4-7	(MSB)	Block Length in Bytes						(LSB)

2.8 Read Format Capacities Command

The READ FORMAT CAPACITIES command allows the host to request a list of the possible capacities that can be formatted on the currently installed medium. If no medium is currently installed, it ***shall*** return the maximum capacity that can be formatted by the device.

Table 2.8 READ FORMAT CAPACITIES Command

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (23h)							
1	Logical Unit Number			Reserved				
2-6	Reserved							
7	Allocation Length (MSB)							
8	Allocation Length (LSB)							
9-11	Reserved							

Table 2.8.1 READ FORMAT CAPACITIES Data Format

Byte	Bit							
	7	6	5	4	3	2	1	0
	Read Format Capacities List Header (see table 2.8.1.1)							
	Current/Maximum Capacity Header (see table 2.8.1.2)							
	Formattable Capacity Descriptor(s) (if any-see table 2.8.1.3)							
0	Formattable Capacity Descriptor 0							
7								
0	Formattable Capacity Descriptor x							
7								

Table 2.8.1.1 READ FORMAT CAPACITIES List Header

Byte	Bit							
	7	6	5	4	3	2	1	0
0-2	Reserved							
3	Capacity List Length							

Table 2.8.1.2 Current/Maximum Capacity Descriptor

Byte	Bit							
	7	6	5	4	3	2	1	0
0	(MSB) Number of Blocks							
1								
2								
3	(LSB)							
4	Reserved						Descriptor Code	
5	(MSB) Block Length							
6								
7	(LSB)							

The Number of Blocks field indicates the number of addressable blocks for the descriptor's media type.

The Descriptor Code field specifies the type of descriptor returned to the host (see 2.8.1.2.1 Descriptor Code Definition).

Table 2.8.1.2.1 Current/Maximum Capacity Descriptor

Descriptor Code	Descriptor Type
00b	Unformatted Media – Other fields invalid
01b	Unformatted Media – Maximum formattable capacity for this cartridge
10b	Formatted Media – Current media capacity
11b	No Cartridge in Drive – Maximum formattable capacity for any cartridge

Table 2.8.1.3 Formattable Capacity Descriptor

Byte	Bit							
	7	6	5	4	3	2	1	0
0	(MSB)							
1				Number of Blocks				
2								
3								(LSB)
4	Reserved							
5	(MSB)							
6				Block Length				
7								(LSB)

The Number of Blocks field indicates the maximum (or fixed) number of addressable blocks for the given capacity descriptor.

The Block Length specifies the length in bytes of each logical block for the given capacity descriptor. There is one format capacity descriptor for each physical format that the device can support.

The device ***shall*** only return Formattable Capacity Descriptors that apply to the installed media. If there is no media installed, the device ***shall*** return a single Formattable Capacity Descriptor, with the maximum capacity that the device is capable of formatting.

The device ***shall not*** return Formattable Capacity Descriptors for formats that it is capable of reading, but not formatting.

If the device is capable of placing more than one physical format on the installed media, the device ***shall*** return one Formattable Capacity Descriptor for each physical format the device can place on the installed media.

2.9 Read Data

The read command requests that the target transfer data to the initiator. The most recent data value written in the addressed logical block shall be returned.

NOTE: This document describes the A8h Read command only. If manufacturers choose to implement target devices supporting the SCSI standard Read(6) and Read(10) byte commands, those devices must accept those commands within a 12 byte wrapper (packet) with all residual bytes padded with a value of zero (0)! In addition, implementing the SCSI Read(6) and Read(10) byte commands must be done in **addition** to the support of the A8h command as described within this document (i.e. it does not negate the requirement of the A8h command).

Table 2.8.1 READ

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (A8h)							
1	Reserved							
2-5	(MSB) Logical Block Address (LSB)							
6-9	(MSB) Transfer Length (LSB)							
10	Reserved							
11	Control							

Logical Block Address field:

The Logical Block Address field specifies the logical block address on the media where the process of reading will begin.

Transfer Length field:

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

2.10 Request Sense

The Request Sense command allows the initiator to request status and sense data from the target device.

The target device shall have valid sense data available if the previous command terminated with a CHECK CONDITION status or if other information is contained in any Sense Data Format field.

If the target device has no valid sense data, it will return a sense key of No Sense and an additional sense code of No Error.

The sense data shall be preserved by the target device until retrieved by a REQUEST SENSE command or until the receipt of another I/O command.

The target device shall report a CHECK CONDITION status associated with the REQUEST SENSE command for fatal errors only.

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

Target devices shall be capable of returning at least 18 bytes of data in response to a REQUEST SENSE command. If the allocation length is 18 or greater, and the target device returns fewer than 18 bytes, the initiator should assume that the bytes not transferred would have been zeros had the target device returned those bytes. An initiator

can determine how much sense data has been returned by examining the allocation length parameter in the REQUEST SENSE command packet and the additional sense length in the sense data. Target devices shall not adjust the additional sense length to reflect truncation if the allocation length is less than the sense data available.

Table 2.9.1 REQUEST SENSE

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (03h)							
1-3	Reserved							
4	Allocation Length							
5-11	Reserved							

The sense data shall be returned in the following format as shown in table 2.9.2 below.

Table 2.9.2 REQUEST SENSE DATA FORMAT

	Bit							
Byte	7	6	5	4	3	2	1	0
0	Valid	Error Code						
1	Reserved							
2	Reserved			ILI	Reserved	Sense Key		
3-6	Information							
7	Additional Sense Length (n-7)							
8-11	Command Specific Information							
12	Additional Sense Code							
13	Additional Sense Code Qualifier							
14	Field Replaceable Unit Code (Optional)							
15	SKSV (Optionl)	Sense Key Specific						
16								
17								
18-n	Additional Sense Bytes							

Valid bit:

A valid bit of zero (0) indicates that the information field is not as defined in this specification. A valid bit of one indicates the information field contains valid information as defined in this specification. The Valid bit shall be implemented for devices adhering to this specification.

Error Code field:

The error code field contains 70h if the sense information relates to the current error. If the sense information relates to a command for which GOOD ending status was previously received, the Error Code field should contain 71h.

ILI bit:

The ILI bit allows the logical device to inform the initiator when the block length requested does not match the length of the logical block residing on the medium. When the ILI bit is set to one, the block length requested does not match the logical block length on the medium.

Sense Key field:

The Sense Key, Additional Sense Code and Additional Sense Code Qualifier provide a hierarchy of information. The intention of the hierarchy is to provide a top-down approach for an initiator to determine information relating to any errors and exception conditions. The Sense Key provides generic categories in which error and exception conditions can be reported. Initiators should typically use sense keys for high-level error-recovery procedures. Additional sense codes provide further detail describing the sense key. Additional sense code qualifiers add further detail to the additional sense code. The additional sense code and additional sense code qualifier can be used by initiators where sophisticated error recovery procedures require detailed information describing the error and exception conditions.

The Sense Key field shall be mandatory and indicates generic information describing an error or exception condition. The sense keys are defined in table 2.9.3 below.

Information field:

The information field is command specific.
Unless specified otherwise this field contains the unsigned logical block address associated with the sense key.

Additional Sense Length field:

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

Command Specific Information field:

This field contains information that depends on the command that was executed. Further meaning for this field is defined within the description of the specific command.

Additional Sense Code field:

The additional sense code field indicates further information related to the error or exception condition reported in the sense key field. Target shall support the additional sense code field. Support of the additional sense codes not explicitly required by this specification is optional. A list of additional sense codes is shown in table 2.9.4 below. If the logical unit does not have further information related to the error or exception condition, the additional sense code is set to NO ADDITIONAL SENSE INFORMATION.

Additional Sense Code Qualifier field:

The Additional Sense Code Qualifier field provides detailed information related to the additional sense code. The additional sense code qualifier is optional. If the error or exception condition is reportable by the logical unit, the value returned shall be as specified in table 2.9.4 below. If the logical unit does not have detailed information related to the error or exception condition, the additional sense code qualifier is set to zero.

Field Replaceable Unit Code field:

Non-zero values in this field are used to define a device-specific mechanism or unit hat has failed. A value of zero in this field shall indicate that no specific mechanism or unit has been identified to have failed or that the data is not available. This field is optional. The format of this information is device specific and not defined in this specification.

SKSV bit:

The SKSV bit indicates when the Sense Key Specific field is valid. A one (1) in this bit validates the Sense Key Specific field. A zero (0) in this field indicates that the value of the Sense Key Specific field is not as defined by this specification. This bit is optional.

Sense Key Specific field:

The value of this field is determined by the value of the sense key field. This field is optional. This field is reserved for sense keys not described in this specification.

Table 2.9.3 SENSE KEY DESCRIPTIONS

Sense Key	Description
0h	NO SENSE: Indicates that there is no specific sense key information to be reported for the designated logical unit. This would be the case for successful command or a command that received CHECK CONDITION or COMMAND TERMINATED status because the ILI bit was set to one.
1h	RECOVERED ERROR. Indicates that the last command completed successfully with some recovery action performed by the target. Details may be determinable by examining the additional sense bytes and the information field. When multiple recovered errors occur during one command, the choice of which error to report (first, last, mode sever, etc.) is device specific.
2h	NOT READY. Indicates that the logical unit addressed cannot be accessed. Operator intervention may be required to correct this condition.
3h	MEDIUM ERROR. Indicates that the command terminated with a non-recovered error condition that was probably caused by a flaw in the medium or an error in the recorded data. This sense key may also be returned if the target is unable to distinguish between a flaw in the medium and a specific hardware failure. (Sense Key 04h).
4h	HARDWARE ERROR. Indicates that the target detected a non-recoverable hardware failure (for example, controller failure, device failure, parity error, etc.) while performing the command or during a self test.
5h	ILLEGAL REQUEST. Indicates that there was an illegal parameter in the command descriptor block or in the additional parameters supplied as data for some commands. If the target detects an invalid parameter in the command descriptor block, then it shall

	terminate the command without altering the medium. If the target detects an invalid parameter in the additional parameters supplied as data, then the target may have already altered the medium. This sense key may also indicate that an invalid IDENTIFY message was received.
6h	UNIT ATTENTION. Indicates that the removable medium may have been changed or he target has been reset.
7h	DATA PROTECT. Indicates that a command that reads or writes the medium was attempted on a block that is protected from this operation. The read or write operation is not performed.
9h	VENDOR-SPECIFIC. This sense key is available for reporting vendor specific conditions.
Bh	ABORTED COMMAND. Indicates that the target aborted the command. The initiator may be able to recover by trying the command again.
Fh	RESERVED.

Table 2.9.4 ADDITIONAL SENSE CODE AND ADDITIONAL SENSE CODE QUALIFIER DESCRIPTIONS
(Numerically ordered by ASC)

Sense Key	ASC	ASCQ	VALID BIT	DESCRIPTION
00h	**00h**	**00h**	**0**	No Error
03h	**01h**	**00h**	**1**	Sector Mark Not Found
04h	**01h**	**81h**	**1**	Sector Pulse Missing.
03h	**03h**	**00h**	**1**	Write Fault
02h	**04h**	**00h**	**0**	Drive Not Ready
02h	**04h**	**01h**	**0**	Drive Becoming Ready
02h	**04h**	**02h**	**0**	Drive Not Ready, Initializing Command Required
02h	**04h**	**03h**	**0**	Intervention Required
02h	**04h**	**04h**	**0**	Not Ready – Format in Progress
04h	**09h**	**00h**	**1**	Track Following Error
01h	**0Ch**	**01h**	**1**	Write Recovered with Sector Reallocation
01h	**0Ch**	**80h**	**1**	Write Recovered via retries
01h	**0Ch**	**81h**	**1**	Write Recovered – PES
01h	**0Ch**	**82h**	**1**	Write Recovered – SM
01h	**0Ch**	**83h**	**1**	Write Recovered – ID
03h	**10h**	**00h**	**1**	ID CRC
03h	**11h**	**00h**	**1**	Unrecovered Read
03h	**12h**	**00h**	**1**	ID AM
03h	**13h**	**00h**	**1**	Data AM
03h	**14h**	**00h**	**1**	Record Not Found
04h	**15h**	**00h**	**1**	Seek Error
04h	**15h**	**01h**	**1**	Seek Too Long
04h	**15h**	**xxh**	**1**	Seek Error (Vendor Specific)
03h	**16h**	**00h**	**1**	Data Sync
01h	**17h**	**01h**	**1**	Data Recovered with Retries
01h	**17h**	**06h**	**1**	Data Recovered with Retries and Sector Reallocation
01h	**18h**	**00h**	**1**	Data Recovered with ECC – No Retries
01h	**18h**	**01h**	**1**	Data Recovered with ECC
01h	**18h**	**02h**	**1**	Data Recovered with ECC and Sector Reallocation
05h	**19h**	**00h**	**0**	Defect List not Ascending
05h	**1Ah**	**00h**	**0**	Parameter List Length Error
00h	**1Ch**	**00h**	**0**	Defect List net Found
01h	**1Ch**	**00h**	**0**	Unsupported Defect List Format
03h	**1Ch**	**00h**	**0**	Defect List Read Error
05h	**20h**	**00h**	**0**	Invalid Command OPCODE
04h	**22h**	**00h**	**0**	Cartridge Present Optical Sensor Calibration Failure
05h	**24h**	**00h**	**0**	Invalid Field in CDB
05h	**25h**	**00**	**0**	Invalid LUN
05h	**26h**	**00h**	**0**	Invalid Field in Parameter List
07h	**27h**	**00h**	**0**	Write Protect
06h	**28h**	**00h**	**0**	Media Changed – Not Ready to Ready Transition
06h	**29h**	**00h**	**0**	Power On or Bus/Device Reset
07h	**30h**	**80h**	**0**	Read Protect
06h	**2Ah**	**00h**	**0**	Mode Select Parameters Changed
05h	**2Ch**	**00h**	**0**	Command Sequence Error
06h	**2Fh**	**00**	**0**	Command Queue Has been Cleared

07h	**30h**	**00h**	**0**	Invalid Media Installed
03h	**31h**	**00h**	**0**	Medium Format Corrupted
03h	**31h**	**01h**	**0**	Format Command Failed
03h	**32h**	**00h**	**1**	No Spare Sectors Available
05h	**39h**	**00h**	**0**	Saving Parameters Not Supported
02h	**3Ah**	**00h**	**0**	Media Not Present
05h	**3Dh**	**00h**	**0**	Invalid Bit in Identify Message
04h	**43h**	**00h**	**1**	Message Failure
0Bh	**43h**	**00h**	**0**	Message Parity Error
04h	**47h**	**00h**	**1**	Parity Error in the Data
0Bh	**47h**	**00h**	**0**	Parity Error in the Command
0Bh	**48h**	**00h**	**0**	Initiator Detected Error Message
0Bh	**4Eh**	**00h**	**0**	Incorrect Initiator Connection Occurred
0Bh	**88h**	**01h**	**1**	Reassigned Block Error
04h	**E1h-E6h**	**00h**	**1**	Vendor Specific

2.11 Seek

The SEEK command requests that the logical unit seek to the specified logical block address.

Table 2.10.1 SEEK

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (2Bh)							
1	Logical Unit Number				Reserved			
2-5	MSB Logical Block Address LSB							
6-8	Reserved							
9	Control							
10-11	Reserved							

Logical Block Address field:

The Logical Block Address to which the operation will seek.

2.12 Send Diagnostics

The SEND DIAGNOSTICS command requests the target to perform diagnostics operations on itself.

Table 2.11.1 SEND DIAGNOSTICS

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (1Dh)							
1	Logical Unit Number			1	Reserved	SelfTest	Reserved	
2-4	MSB Parameter List Length LSB							
5	Control							
6-11	Reserved							

SelfTest bit:

A self-test bit of one (1) directs the target to complete its default self-test. If the self-test successfully passes, the command shall be terminated with GOOD status; otherwise, the command shall be terminated with CHECK CONDTION status and the sense key shall be set to HARDWARE ERROR.

A self-test bit of zero (0) requests that the target perform the diagnostics operation specified in the parameter list. The diagnostic operation might or might not require a target to return data that contains diagnostic results. If the return of data is not required, the return of GOOD status indicates successful completion of the diagnostic operation. If the return of data is required, the target shall either:

A) Perform the requested diagnostic operation, prepare the data to be returned and indicate completion by returning GOOD status. The initiator issues a RECEIVE DIAGNOSTIC RESULTS command to recover the data.

B) Accept the parameter list, and if no errors are detected in the parameter list, return GOOD status. The requested diagnostic operation and the preparation of the data to be returned are performed upon receipt of the RECEIVE DIAGNOSTIC RESULTS command.

Parameter List Length field:

The parameter list length field specifies the length in bytes of the parameter list that shall be transferred from the initiator to the target. A parameter list length of zero indicates that no data shall be transferred. This condition shall not be considered an error.

2.13 Start/Stop Unit

The START/STOP UNIT command requests that the logical unit enable or disable media access.

Table 2.12.1 START/STOP UNIT

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (1Bh)							
1	Reserved							Immed
2-3	Reserved							
4	Reserved						LoEj	Start
5	Control							
6-11	Reserved							

Immed bit:

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

Start bit:

Refer to table 2.6.2 for a description of this bit.

LoEj bit:

Refer to table 2.6.2 for a description of this bit

2.13 Test Unit Ready

The TEST UNIT READY command tests the readiness for media access of the target device. This shall not be a request for self-test. If the logical unit is able to accept an appropriate medium-access command without returning CHECK CONDITION status, this command shall return GOOD STATUS. If the logical unit is unable to become operational, or is in a state such that initiator action is required to make the unit ready, the target shall return CHECK CONDITION status with a sense key of NOT READY.

Table 2.7.1 TEST UNIT READY

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (00h)							
1	Logical Unit Number			Reserved				
2-4	Reserved							
5	Control							
6-11	Reserved							

2.14 Verify

TBD

2.15 Write and Verify

TBD

2.16 Write Data

TBD

4.0 Removable Media Support

4.1 Introduction

There are two methods for support of Removable Media provided in this specification. These two methods are MSN and MESN. It is up to the discretion of the manufacturer of the target device which of the two (or both) shall be implemented for any particular target device or set of target devices.

MSN shall be used for ATA-4 compliant devices (ATAPI). MESM may be used for all Removable Media devices (including ATAPI if desired) that implement command protocol in a packetized manner (ATAPI, USB, 1394. etc.).

For devices choosing to implement MESN, this functionality must be indicated in the INQUIRY command return information, bit 4 byte 3. See the INQUIRY command section 2.3 for more information.

For devices choosing to implement MSN, this functionality must be indicated in the IDENTIFY command return data, byte 127, bits 1-0. Refer to Specification SFF-8070 ATAPI Removable Rewritable Media for more information.

Both methods shall be discussed in this specification.

4.2 MSN

The MSN feature in this specification is patterned directly after the ATAPI Removable Rewritable Media SFF-8070 specification. Refer to this specification for more detailed information regarding MSN functionality.

Returning Media Status

The media status information is returned when the host issues the GET MEDIA STATUS (0xDA) command. The command protocol is as follows:

1. GET MEDIA STATUS (0xDA) command is written to the Command Register.
2. ATAPI Block Device asserts BSY.
3. After ATAPI Block Device has completed processing the command, it clears BSY and sets INTRQ.
4. The host reads the Status Register.
5. ATAPI Block Device clears INTRQ.

If the state of ATAPI Block Device is normal (that is, medium in ATAPI Block Device, medium not changed, Eject button not pushed, medium not read- or write-protected), the GET MEDIA STATUS command must return good status. If the command fails due to a hardware fault, the command must be failed with the ERROR bit set in the status register, and the ABORT bit set in the error register.

If the state of ATAPI Block Device has changed (or no medium is in ATAPI Block Device), the GET MEDIA STATUS command must be failed with the ERROR bit set in the status register, and the appropriate bits set in the ERROR register as follows.

Table 4.2.1 ERROR REGISTER

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RD_PT	WRT_PT	MC	RSVD	MCR	RSRVD	NOMED	RSRVD

These bits are defined as follows:

RSRVD

This bit ***shall*** be zero.

NOMED

This bit indicates there is currently no medium in ATAPI Block Device. This bit must be set for each invocation of the GET MEDIA STATUS command until the user inserts a medium in ATAPI Block Device.

MCR

This bit indicates that the user has attempted to eject the medium. Usually this bit will be set when the user presses the Eject button on ATAPI Block Device. If ATAPI Block Device is capable of preventing removal of the media, ATAPI Block Device ***shall not*** allow the medium to be ejected until an EJECT command is received from the host. This bit ***shall not*** be set again in response to the GET MEDIA STATUS command until the user subsequently releases the button and presses it again. ATAPI Block Device must take no action with regard to the user's attempt to eject the medium other than setting this bit, except in the case that the ELECT_EJ functionality is not reported in the CYL HIGH register from the ENABLE MEDIA STATUS SET FEATURES command, which means that ATAPI Block Device cannot control when the medium is ejected.

MC

This bit indicates that a medium has been newly inserted in ATAPI Block Device. The bit must not be set for more than one invocation of the GET MEDIA STATUS command, until the user removes or reinserts the medium. In other words, when the user inserts a new medium in ATAPI Block Device, and a GET MEDIA STATUS command is issued, ATAPI Block Device will set this bit. The next GET MEDIA STATUS command must not result in this bit being set again unless the medium has been removed and reinserted.

WRT_PT

This bit indicates that a medium is currently write protected.

RD_PT

This bit indicates that a medium is currently read protected.

Performance Notes:

Get Media Status. The GET MEDIA STATUS command must be completed within the next I/O cycle (or as soon as possible) in order to facilitate drivers that, for performance reasons, poll BSY rather than waiting for an interrupt. Therefore, ATAPI Block Device's firmware implementation might need to determine ATAPI Block Device status periodically, rather than having to determine it, if necessary, when the GET MEDIA STATUS command is issued.

ATA Compatibility Considerations

Lock/Unlock/Eject. The LOCK and UNLOCK ATA commands or the equivalent PREVENT/ALLOW ATAPI commands have no function when Media Status Notification support is enabled. Any Lock or Prevent that was in effect when Media Status Notification is first initiated will be cleared. If supported, these commands must be completed, but must have no effect on the state of ATAPI Block Device. The EJECT command must always eject the medium, regardless of any pending lock commands.

Acknowledge Media Change. The ACKNOWLEDGE MEDIA CHANGE ATA command must not be required when Media Status Notification support is enabled. However, if the host issues the command, it must be completed without error if supported. Note that this command is obsolete and is no longer supported in ATA-3. ATA drives must fail any READ, WRITE, VERIFY, LOCK, UNLOCK, or EJECT command when Media Status Notification support is enabled and no medium is in ATAPI Block Device. Also, as defined in the "Returning Media Status" section earlier in this specification, the NOMED bit must be the error reported if these commands are received with no medium. This behavior differs from that of some currently available ATA removable devices, which will "virtualize" certain commands and return "fake" data when no medium is in ATAPI Block Device. A problem arises if ATAPI Block Device wants to report the TK0NF error, because the NOMED bit replaces this error bit when Media Status Notification is enabled. This type of error should be reported using one of the other ERROR bits.

Also, the ATAPI Block Device must return the status change in the ERROR register just as it is defined for the GET MEDIA STATUS command if the following two conditions occur:

1. Media Status Notification is enabled and the state of ATAPI Block Device changes (that is, the user inserts a medium or presses a button)
2. an ATA command other than the GET MEDIA STATUS command is received by ATAPI Block Device before the GET MEDIA STATUS command is received

ATAPI Block Device must also clear the condition just as defined for the GET MEDIA STATUS command. Only changes in the state of ATAPI Block Device with respect to the bits MC, MCR, and NOMED (as defined in the GET MEDIA STATUS command) must be handled in this manner. The READ PROTECT and WRITE PROTECT bits, of course, must not be reported as errors except from the GET MEDIA STATUS command or READ or WRITE requests, respectively.

ATAPI Compatibility Considerations

ATAPI commands must be processed exactly the same as they would be if Media Status Notification was not enabled. For example, CHECK CONDITION with UNIT ATTENTION must be returned for ATAPI commands under the same conditions that would cause the MC bit to be set.

When the GET MEDIA STATUS command is received by the ATAPI Block Device, the MCR, NOMED, MC, WRT_PT, and RD_PT bits must be reported as described in "Returning Media Status" earlier in this specification. However, the state of the bits must not be cleared by any ATAPI command. The insertion of a new medium must be reported independently with both the GET MEDIA STATUS and ATAPI commands. For example, if the user inserts a new medium and ATAPI Block Device is accessed with an ATAPI command, the CHECK CONDITION with UNIT ATTENTION must be reported. However, ATAPI Block Device must also report the MC error when it receives the next GET MEDIA STATUS command. The behavior must be the same if, after the medium is inserted, the GET MEDIA STATUS command is received before the next ATAPI command.

When the user presses the Eject button on ATAPI Block Device, the ATAPI commands must not be affected. However, the MCR status must be reported when the first GET MEDIA STATUS command is received after the button is pressed (it is reported only once after each button press). The ATAPI Block Device must continue to operate as normal. If the operating system determines that it is safe to eject the medium, an ATAPI EJECT command must be issued to ATAPI Block Device.

ATAPI Block devices that support multiple LUN's ***shall*** honor the LUN field in the ATAPI Block Device Select register for the GET MEDIA STATUS command. For example, if the medium is inserted in LUN 2 of a device, a GET MEDIA STATUS command issued for LUN 2 ***shall*** report the change, but a GET MEDIA STATUS command for LUN 0 ***shall not***. The ENABLE and DISABLE MEDIA STATUS SET FEATURES commands are not LUN based. These commands ***shall*** be applicable for the entire device, and ***shall*** enable Media Status Notification on all LUN's present. The protocol for reporting and accessing multiple LUN's is defined in SFF-8020i Revision 2.6 and higher.

4.3 MESN

4.3.1 Event Status Notification

As it is difficult to asynchronously interrupt the host due to lack of industry support for Asynchronous Event Notification, the GET EVENT STATUS NOTICATION (GESN) and Event Status Notification feature provide a method for the host to receive notification of events that are beyond host control.

A logical unit that implements Event Notification shall support the command as specified in table 4.3.1.1 below.

Table 4.3.1.1 Media Event Status Notification Feature Set

Command Description	Opcode
GET EVENT STATUS NOTIFICATION	4Ah
PREVENT/ALLOW MEDIA REMOVAL with PERSISTENT PREVENT option	1Eh

Under GESN, the host shall repeatedly issue GESN commands with an immediate bit of 1. The logical unit shall complete these commands upon receipt, supplying the host with information on the most recent event occurrences, as described in the GESN command (See section 3.3.4 below). If an event occurrence of the class(es) requested is not in the logical unit event queue, the logical unit shall complete the GESN command, and shall set the NEA bit to 1. This shall not be deemed and error.

If command queuing is supported, the host may issue a GESN command with an immediate bit of 0. The command shall not complete until an event occurrence of the class(es) requested is either in the vent queue, or occurs.

The logical unit shall maintain a separate queue for each class of Event Notification(s) supported. Events that are generated shall be placed at the tail of the event queue. The depth of the queue(s) is vendor specific, although is shall be at least one. If an overflow occurs, the logical unit shall maintain the most recent Events in the queue.

Each GESN command shall report only one event. If multiple Event Classes are requested and multiple events are available, the logical unit shall report the Event in the Event Class with the lowest Notification Class ordinal.

It should be noted that a major shortcoming of removable media logical units on PC platforms is their inability to report to the host when the user attempts to eject the medium. Currently most removable media logical units just eject the medium when the user presses the eject button, and potentially any data the operating system has not saved to the logical unit is lost. Various volume tracking and locking schemes reduce this risk, but none eliminate it. Using the method described in this specification, logical units will have a means of communicating to the host that the user want to eject the medium or has inserted a new medium.

This section defines a protocol for providing this functionality for removable media logical units. The support is enabled using a new bit contained in the PREVENT/ALLOW MEDIA REMOVAL command (Persistent Bit) (See section 2.6 for more detail on the PREVENT/ALLOW MEDIA REMOVAL command).

Once the support is enabled, the media status may then be retrieved using the GET EVENT STATUS NOTIFICATION command as described below.

When the Persistent Prevent state is entered (by setting the Persist bit of the PREVENT/ALLOW MEDIA REMOVAL command), the media shall remain “locked” in the logical unit until the host issues and eject request, or a power on or hard reset condition occurs. The Persistent Prevent state shall be maintained after the eject request. New media that is inserted into the logical unit shall be

locked in the logical unit after the logical unit reports the NEW MEDIA event. Prior to reporting the NEW MEDIA event, the logical unit may eject media without an explicit eject command from the host. This allows the user to remove incorrectly inserted media without having to wait for host intervention.

While in the Persistent Prevent state, the logical unit shall generate Events upon receipt of a User Eject request. The logical unit shall not eject the media on receipt of these requests if the logical unit has already reported a NEW MEDIA event for this media. When the host receives the Eject Request, and determines that it is safe to eject the medium, an eject command will be issued, at which time the logical unit shall eject the medium.

The logical unit shall only generate MESN (EJECT REQUEST) events after reporting a MESN (NEW MEDIA) event, and prior to reporting a MESN (MEDIA REMOVAL) event for the given media.

NOTE: Refer to the definition of the Get Event Status Notification command (see section 4.4) for information regarding the meanings of the EJECT REQUEST, NEW MEDIA and MEDIA REMOVAL MESN event notifications.

4.3.2 Compatibility Considerations

To maintain compatibility with existing BIOS implementations and operating systems, the logical unit shall default to the Persistent Prevent disabled state. When the host enables the support using the PREVENT/ALLOW MEDIA REMOVAL command, the logical unit shall respond as described in this specification. When the host disables this feature, the logical unit must default to normal operating modes. A power on or hard reset shall cause the logical unit to the default Persistent Prevent disabled state.

If the logical unit is unable to maintain media status information across a reset or power cycle, the logical unit shall generate a NEW MEDIA event.

Commands must be processed exactly the same as they would be if Persistent Prevent was not enabled. For compatibility reasons, UNIT ATTENTION status conditions must still be returned. However, the logical unit shall not return the UNIT ATTENTION status on a GESN command. For example, if the user inserts a new medium and the logical unit is accessed with a command, the CHECK CONDITION with UNIT ATTENTION shall be reported, but the logical unit shall also report the NEW MEDIA event with the next available GESN command.

4.4 Get Event Status Notification Command

The Get Event Status Notification Command requests the logical unit to report event(s) status(es) as specified in the Notification Class field and provides asynchronous notification. Two modes of operation are defined here. They are polling and asynchronous modes.

In polling mode, the host will issue Get Event Status Notification commands at periodic intervals with an IMMED (immediate) bit set to 1. The target shall complete this command with the most recently available event status requested. The support of this mode is mandatory.

In asynchronous mode, the host will issue a single Get Event Status command with an IMMED (immediate) bit set to 0. If the logical unit supports asynchronous event status notification (through tagged queuing), the model outlined here shall be used. If the logical unit does not support asynchronous mode, the command shall fail as an illegal request.

When asynchronous Event Status reporting is supported, the target shall not complete a Get Event Status command with an IMMED bit of 0, until a change in event status of the requested class occurs. The target shall complete the Get Event Status Notification Command as soon after the event occurs as possible. It will report the event as outlined below.

NOTE: Only one call of event per Get Event Status Notification Command shall be reported. The priority of event reporting shall be by event Class number. The lower the class number, the higher the priority.

This command shall not return a Unit Attention check condition.

Table 4.4.1 GET EVENT STATUS NOTIFICATION COMMAND

	Bit							
Byte	7	6	5	4	3	2	1	0
0	OPCODE (4Ah)							
1	Logical Unit Number			Reserved				Immed
2-3	Reserved							
4	Notification on Class Request							
5-6	Reserved							
7	Event List Length (MSB)							
8	Event List Length (LSB)							
9	Control							
10-11	Pad (0)							

The Notification Class Request field requests the logical unit to report event(s) from the list of available classes as shown in table 3.3.3.2 below.

Table 4.4.2 NOTIFICATION CLASS DEFINITIONS

Bit	Definition
0	Reserved
1	Reserved
2	Power Management Class Events
3	Reserved
4	Media Status Class Events
5	Reserved
6	Device Busy Class Events
7	Reserved

NOTE: A bit field of all 0s indicates that the target should immediately complete this command indicating “No Event,” and shall list the supported event class in the

Event Buffer header. This method shall be used to determine which event classes a logical unit supports.

If a logical unit does not support any of the requested event classes, the logical unit shall terminate the command successfully, returning only the Event Data Header, and indicating a returned class of 0.

Host Software that manages media event status, may or may not be linked to other software that manages power states. This notification field provides a way that power and media event status notifications can be independently managed by the responsible software. If a driver manages media, power management and Busy Device events, the driver can issue this command with notification field set to 0101010b to request the logical unit to report both power and media events.

The information sent from the logical unit to the initiator or host shall conform to the following format as specified in table 4.4.3 below.

Table 4.4.3 Notification Status Format

	Bit							
Byte	7	6	5	4	3	2	1	0
0-n	Event Header							
0-n	Event Data							

The Event Header format is shown below in table 4.4.4.

Table 4.4.4 Event Status Header Format

	Bit							
Byte	7	6	5	4	3	2	1	0
0	MSB Event Data Length							
1								LSB
2	NEA	Reserved				Notification Class		
3	Supported Event Classes							
4-n	Class Event Data							

Event Data Length field:

The Event Data Length field specifies the amount of data that follows the Event Status Notification Header. The amount of data reported shall be the number of bytes of data following the data length field.

NEA bit:

This bit indicates that there are events available in the requested Notification Class(es). It this bit is set, the information is available. If this bit is not set, not information about the requested event class is available.

Notification Class field:

This field specifies the class of notification by number. The supported values are shown in table 4.4.5 below.

Table 4.4.5 Notification Classes

Field	Description
000b	None of the requested event classes are supported
001b	Reserved
010b	Returns Power Management class events
011b	Reserved
100b	Returns Media class events
101b	Reserved
110b	Device Busy class events
111b	Reserved

The following tables describe the individual Classes listed in table 3.3.3.6.

Table 4.4.6 Power Management Status Class Return Data

Byte	7	6	5	4	3	2	1	0
0	Reserved				Power Event			
1	Power Status							
2	Reserved							
3	Reserved							

Power Event field:

The Power Event field reports the current change in the power status. This field is set to a new power event if a change in the power state occurs.

Upon reporting the current power status change to the host, this field is reported as 0h on subsequent Get Event Status Notification commands until a new change in power state occurs.

If the logical unit is commanded to enter the state that the logical unit is currently in, the next Get Event Status Notification (Power Class) command shall report a Power Change Successful event.

Table 4.4.6.1 Power Event Field Format

Code	Event	Description
0h	NoChg	No changes in power state, or in power state transition
1h	PwrChg-Succ	The logical unit successfully changed to the specified power state
2h	PwrChg-Fail	The logical unit failed to enter the last requested state, and is still operating at the power state specified in the Power Status field.
3h-Fh	Reserved	

Power Status field:

The power status field indicates the current power status of the logical unit. The possible values for the power status field are shown in table 4.4.6.2 below.

Table 4.4.6.2 Power Status Field Format

Code	Status	Description
0h	Reserved	
1h	Active	The logical unit is in the Active State
2h	Idle	The logical unit is in the Idle state
3h	Standby	The logical unit is in the Standby state
4h-Fh	Reserved	

The Media Event Class Return Data Format is described below.

Table 4.4.7 Media Event Class Return Data

Byte	7	6	5	4	3	2	1	0
0	Reserved				Media Event			
1	Media Status							
2	Start Slot							
3	End Slot							

Media Event field:

The supported values for the Media Event field are shown in table 4.4.7.1 below.

Table 4.4.7.1 Media Event Format

Code	Event	Description
0h	No Event	Media status is unchanged
1h	EjectRequest	The logical unit has received a request from the user (usually through a mechanical switch on the logical unit) to eject the specified slot or media
2h	NewMedia	The logical unit (or specified slot) has received new media, and is ready to access it.
3h	MediaRemoval	The media has been removed from the specified slot, and the target is unable to access the media without user intervention.
4h-Fh	Reserved	

Media Status field:

The Media Status field (1 byte) is shown in table 4.4.7.2 below.

Table 4.4.7.2 Media Status Byte Format

Byte	7	6	5	4	3	2	1	0
1	Reserved						Media Present	Door open

Media Present bit:

The Media Present bit indicates the presence of media in the logical unit. A value of one (1) in this bit indicates that there is media present in the device. A zero (0) in this bit indicates that no media is present.

This bit is reported independently of the Door open bit. If the logical unit does not support the capability of reporting the media state while the door is open, shall set the Door open bit to zero (0).

Door open bit:

The open bit indicates (for logical units that detect the state of a media door) the status of the device media door. A one (1) indicates that the door is open. A zero (0) in this bit indicates that the door is closed.

For devices that do not support the detection of the state of the media door, this bit shall be set to zero (0).

Start Slot field:

The Start Slot field defines the first slot of a multiple slot logical unit to which the media status notification applies. For logical units that do not support multiple slots, this field shall be reserved.

End Slot field:

The End slot field defines the last slot of a multiple slot logical unit to which the media status notification applies. For logical units that do not support multiple slots, this field shall be reserved.

Table 4.4.8 Device Busy Event Class Return Data

Byte	7	6	5	4	3	2	1	0
0	Reserved				Device Busy Event			
1	Device Busy Status							
2	MSB Time							
								LSB

Device Busy Event field:

The supported Device Busy Event values are shown in table 4.4.8.1 below.

Table 4.4.8.1 Device Busy Event Values

Code	Event	Description
0h	NoEvent	No event is available
1h	Busy Event	A time-out has occurred
2h-Fh	Reserved	

Table 4.4.8.2 Device Busy Status Values

Code	Event	Description
0h	NoEvent	Logical unit is ready to accept any command
1h	Power	The logical unit is in the process of waking up from a low power state.
2h	Immediate	The logical unit is in the process of completing an earlier command
3h	Deferred	The logical unit is in the process of completing a deferred operation, such as a write.
4h-FFh	Reserved	

5.0 Prevent/Allow Media Removal

Media Event Status Notification support is enabled via a new bit defined in the Prevent/Allow Media Removal command. This new bit is the Persist bit.
Refer to section 2.6 for a description of the Prevent/Allow Media Removal Command.