X3T9.2/92-56

adaptec, inc.

PRELIMINARY
X3T9.2 Proposal for SCSI-3

TO: John Lohmeyer, Chairman of X3T9.2 SCSI Committee

FROM: Al Wilhelm **DATE:** March 12, 1992

SUBJECT: Proposal to modify Device Type 0 to support Media Interchange "Revised"

On May 15, 1991, at the working group meeting in Harrisburg I submitted a proposal requesting a new device type for High Capacity SCSI Floppy drives. This would be a very 'clean' solution to solving the existing problems in providing for 'media interchange,' the ability to have a drive create 100% compatible media regardless of the SCSI Host Adapter (SHA) to which it is attached. It would also make it easy to solve the interface problem to the user (Floppy vs hard disk). The committee made it clear that they were **against the addition of a new device type unless it could be shown that support of media interchange was impossible within the existing device type 0, Direct Access Storage Device (DASD).** In keeping with the committee's direction, this proposal suggests changes to the command set for device type 0 to support media interchange and to prevent the creation of incompatible media on a 'Media Interchange' device.

[Aside: There are still several experienced software engineers who feel strongly that a new device type is both justified and the best solution to this problem.]

This proposal's intent is to address the global issue of media interchange within the existing device type 0. Please refer to **Attachment A** for a partial description of the existing media interchange problem. Throughout this document I use **SHA** to refer to **SCSI Host Adapter** boards and associated software drivers from any company. All SCSI references are to SCSI-2 spec X3T9.2/86-109 **Rev 10h**.

First let me define a new subtype within the existing SCSI device type 0. I will refer to this new subtype as a **Media Interchange or MI device.** An MI device adheres to the changes described in this proposal. An MI device supports certain media types and uses a standard format for each media type. Any media created by an MI device will be fully interchangeable with any other MI drive that can handle the same media type -- regardless of the SHA to which it is attached.

[An example of an MI device is the Floptical™ drive manufactured by both Insite and IOMega. In addition to Reading/Writing the common 3½" 1M and 2M Byte (720k and 1.44M Byte formatted) diskettes that are used in PCs throughout the world, it supports a new 20+ MByte 3½" diskette (20.8 MBytes formatted) with a new standard format. Japanese makers of SCSI floppy drives that adhere to the JEIDA formats could be another example. Also, it would be possible for new removable disk drives to utilize this scheme and provide for more universal media interchange as well.]

THERE ARE ESSENTIALLY TWO GOALS:

GOAL 1: Provide a method by which an "MI aware" SHA can distinguish an MI drive from the non-MI drives and identify which media types and media formats are supported. Only industry standard media formats are generated -- the same format per media type, regardless of SHA used.

GOAL 2: Ensure that creation of incompatible media is totally prevented. To do this it is necessary to ensure that any older or non-MI aware SHA can not write, format, or alter the data on an MI drive's media.

This proposal attempts to satisfy those goals in as simple a way as possible. It recommends changes to only the INQUIRY and START STOP UNIT commands, and a new MODE PAGE 06. It incorporates the Density Codes as defined and used by Sequential-access devices into the Direct-access device command set. It also suggests how a "well behaved" MI drive functions.

INQUIRY Command I suggest redefining bit 5 of byte 3 of the Standard INQUIRY Data Format to be the **MI bit.** All existing drives and future non-MI drives leave this bit cleared as before. All MI drives set this bit to 1. An "MI-aware" SHA can now easily distinguish an MI drive from a non-MI drive.

TEST UNIT READY, FORMAT UNIT, and WRITE Command No changes are made to the TUR, FORMAT UNIT and/or WRITE commands. An MI drive will respond to a TUR, FORMAT UNIT, or WRITE command with STATUS 02, CHECK CONDITION. It responds to a subsequent REQUEST SENSE Command with SENSE KEY 02, NOT READY; ASC ___, MI DRIVE REQUIRES START COMMAND. New ASC number should be assigned and added to "Table 71: ASC and ASCQ Assignments." An 'MI Aware' SHA should interpret this to mean that a START STOP UNIT Command is required as defined in the next paragraph. [An older non-MI aware Host Adapter may send a START STOP UNIT Command with the START bit set (bit 0 of byte 4). Although adequate for existing non-MI drives, this will NOT initialize an MI drive!]

START STOP UNIT Command I suggest redefining bit 1 of byte 1 of the START STOP UNIT Command to be the IMI (Initialize MI) bit. Only an 'MI-aware' SHA knows to set this bit. An MI drive must receive a START STOP UNIT Command with both the IMI bit and the START bit set in order to be properly initialized. Until this happens, an MI drive can do either of the following: (1) Return Check Condition to all commands except INQUIRY and MODE SENSE or (2) Return Check Condition to only commands that can alter the media (i.e. FORMAT UNIT, WRITE, etc.) It is up to the drive manufacturer to decide which commands will be "locked out" until the drive is properly initialized by an MI aware SHA as long as alteration of the media is absolutely prevented. I refer to all commands that are rejected until the MI drive is properly initialized (or unlocked) as the "Locked Out Commands."

Once the MI drive is properly initialized by an MI aware SHA the MI drive is fully functional and responds to all commands as a good MI drive should.

WHAT IF THERE IS NO MEDIA IN THE DRIVE? Until the MI drive has been properly initialized (or unlocked), it responds to all 'Locked Out Commands' with CHECK CONDITION and then SENSE KEY 02, ASC ___ , MI DRIVE REQUIRES START COMMMAND, as described above. Once unlocked, the MI drive responds to any command as described in the SCSI-2 spec rev 10h. For example: If there is no media inserted, it might respond to a TUR, WRITE, READ, etc. with CHECK CONDITION and then SENSE KEY 02, NOT READY; and ASC 3A, MEDIUM NOT PRESENT as is already defined in Tables 69 and 71.

HOW DOES THE HOST ADAPTER KNOW WHAT MEDIA TYPES ARE SUPPORTED?
This is already defined in the spec. MODE SENSE Page 0B is currently defined to indicate up to four media types supported. These medium types are not to be confused with medium formats (although there is often a 1-to-1 correlation). They are defined in **Table 153: Direct-Access Medium-Type Codes.**
[Aside: It is unfortunate that this page was defined to report only four types as this will not stand the test of time. Drives that support more than four types are just around the corner (consider the existing 3.5" 1M and 2M Byte diskettes, add the new 4M Byte and 21M Byte diskette types soon to be followed by a 40M Byte flavor.) This mode page is OK for now but will need to be expanded in the future, maybe to variable length.]

HOW DOES THE SHA KNOW WHAT MEDIA IS CURRENTLY INSERTED? That is already in the spec as well. The Mode Parameter Header gives the medium type that is currently installed in the drive. For a 6-byte MODE SENSE Command, Medium Type is reported in byte 1 of the Mode Parameter Header (Table 91); for a 10-byte MODE SENSE Command, Medium Type is reported in byte 2 of the Mode Parameter Header (Table 92).

DIRECT-ACCESS DEVICES SHOULD USE "DENSITY CODES"
AS DEFINED BY SEQUENTIAL-ACCESS DEVICES

In section 8.3.3 I recommend replacing the text relating to the density code field in the mode parameter block descriptor with text similar to that found in section 9.3.3.

Existing text in section 8.3.3 (top sentence of page 224):
The density code field is contained in the mode parameter block descriptor (see 7.3.3). This field is reserved for direct-access devices.

I propose changing 8.3.3 by replacing the above sentence with the following text (mostly from 9.3.3 pages 285, 286):
For the MODE SELECT command, the density code field of the direct-access device block descriptor (see 7.3.3) indicates the density selected by the initiator for use in subsequent read and write operations. For devices capable of automatic density recognition, the density code selected by the initiator may be overridden by the target for a subsequent read operation if the selected value does not match the current recorded density of the medium. If the MODE SELECT command specifies the default density code the device selects the actual density code to be used in a vendor specific manner. The value is expected to be the principal density code (or an optimal density code).

For the MODE SENSE command, the density code field reflects the current operating density of the device. If a current operating density has not been selected, either because no medium is installed or because the density of the installed medium has not been determined, the density code field should be set to the principal density code value. The principal density code is the highest density code supported, the optimal density code supported, or the most common density code supported. In some devices, the principal density code value returned in response to a MODE SENSE command may change dynamically to match the most recently selected density. The density code value returned in response to a MODE SENSE command shall be as described below:

1) Following a UNIT ATTENTION condition for a power on or hard reset condition, while not ready, the target shall report the principal density.
2) Following a UNIT ATTENTION condition for a not-ready-to-ready transition, the target shall:
   a) report the principal density if no attempt has been made by the target to determine the density.
   b) report the principal density if the target cannot automatically determine the density from the medium.
   c) report the current medium density if the target can determine the density from the medium.
3) Following a successful read operation, the target shall report a density code value reflecting the recorded density of the medium. For some implementations, the target may automatically determine this value from the medium. For devices not capable of automatic density determination, the principal density is reported if the density code value is not provided by the preceding MODE SELECT command.
4) Following an unsuccessful read operation or a successful write operation, the target shall:
   a) report a density code value as described for item (2) if a previous MODE SELECT command has not established a density code for the currently mounted medium.
   b) report a density code value as provided by the last successful MODE SELECT command for the currently mounted medium.
5) Following dismount and/or removal of medium, the target shall report the most recent density code value as determined by items (2) through (4) above.

Table Y lists the direct-access device density codes.

Table Y: Direct-Access Density Codes

Code Value	Description
00h	Default Density (MODE SELECT command only)
01h - 7Eh	Reserved
7Fh	No change from previous density (NO-OP) (MODE SELECT command only)
80h - FFh	Vendor Specific

[The above line marks the end of the suggested text about Density Codes to be added to 8.3.3]

Note: As Disk/Diskette drive makers begin to use Density Codes for the Medium Types used by new drives that they develop, they should propose both a Density Code number and associated definition for inclusion into Table Y.

Existing Mode Page 05 – for reference only For the reader's convenience I have duplicated the existing (and possibly useless) Mode Page 05h "Table 158: Flexible Disk Page" from the SCSI spec. On the next page, I show my suggested new page 06h "Table X: Medium Interchange Disk Page."

The reader should also refer to MODE PAGE 04h "Table 171: Rigid Disk Drive Geometry Page." Those physical disk parameters are useful for specialized non-DOS driver software that is optimized for particular disk drives (e.g. for disk array striping).

Table 158: Flexible Disk Page

\ Bit / Byte\	7	6	5	4	3	2	1	0	
0	PS	Reserved		Page Code (05h)					**Existing Page**
1				Page Length in Bytes (1Eh)					
2 *	(MSB)			Transfer Rate *					
3 *								(LSB)	
4				Number of Heads					\ Physical Floppy
5				Sectors per Track					> Diskette
6	(MSB)			Data Bytes per Sector					/ Parameters
7								(LSB)	/
8	(MSB)			Number of Cylinders					
9								(LSB)	
10 *	(MSB)			Starting Cylinder-Write Precompensation *					
11 *								(LSB)	
12 *	(MSB)			Starting Cylinder-Reduced Write Current *					
13 *								(LSB)	
14 *	(MSB)			Drive Step Rate *					
15 *								(LSB)	
16 *				Drive Step Pulse Width *					
17 *	(MSB)			Head Settle Delay *					
18 *								(LSB)	
19 *				Motor On Delay *					
20 *				Motor Off Delay *					
21 *	TRDY*	SSN *	MO *	Reserved					
22 *	Reserved						SPC *		
23 *				Write Compensation *					
24 *				Head Load Delay *					
25 *				Head Unload Delay *					
26 *	Pin 34 *						Pin 2 *		
27 *	Pin 4 *						Pin 1 *		
28 *	(MSB)			Medium Rotation Rate *					
29 *								(LSB)	
30 - 31				Reserved					

* These entries may have been intended for SCSI-to-Floppy bridge controllers. They seem mostly useless for an embedded SCSI floppy drive and will probably never be used anymore if they ever were. Please inform me if my assumption in this matter is incorrect. In any event, I am not making changes to Mode Page 05h.

4/10

Proposed New MODE PAGE 06h – Medium Interchange Disk Page.

I believe MODE PAGE 06h is currently reserved for future use. Here is my suggestion for a new MODE PAGE 06h.

Table X: Medium Interchange Disk Page

\ Bit | Byte\	7	6	5	4	3	2	1	0
0	PS	Reserved	Page Code (06h)					
1	Page Length in Bytes (1Eh)							
2 - 3	Reserved							
4	Number of Logical Heads							
5	Logical Sectors per Track							
6	(MSB)	Data Bytes per Sector						
7								(LSB)
8	(MSB)	Logical Number of Cylinders						
9								(LSB)
10	PART	Reserved –					EJCT	NHF
11 - 24	Vendor Unique							
25 - 27	Reserved							
28 - 29	Vendor Unique							
30 - 31	Reserved							

Proposed New Page for Medium Interchange Drive (both floppy and/or removable hard disk).

\ Logical
> Disk
/ Parameters
/

Logical Parameters are used by every SHA for consistent translation scheme needed to interface to various OS' used in PC/AT machines.

[Here is the suggested text to follow Table X]

The Medium Interchange Disk Page (Table X) contains parameters for control and reporting of Medium Interchange Disk Drive Parameters. These parameters are to be used by an MI-aware SCSI Host Adapter for the translation table that they use to interface to the Host Operating System (if necessary) in order to create media that is compatible to media created by other MI aware SCSI Host Adapter implementations. The Logical Parameters refer to the media type that is currently inserted in the drive.

[Text for the PS bit stays the same. In general, this page will probably not be savable.]

This page is 32 bytes in length. Bytes 0-32 are defined in Table X.

The Logical Heads, Logical Sectors per Track, Bytes per sector, and Logical Cylinders fields are the HAM (Host Adapter Manufacturer) format recommended by this drive for the media type that is currently inserted. All MI aware SCSI Host Adapter implementations shall use these parameters for the 'physical-to-logical' translation table that they use (if needed) to interface to the Host Operating System.

The PART bit indicates whether this drive is to be treated as a partitionable drive (i.e. removable hard disk) or non-partitionable (i.e. floppy format). A PART bit of one indicates that the drive should be treated as a partitionable device. A PART bit of zero indicates that the drive should be treated as a floppy device and should not be partitioned.

The EJCT bit is used to indicate whether or not this drive supports ejection of the media. An EJCT bit of one indicates that this drive will eject media when it receives a START STOP UNIT Command with the LoEj bit set. An EJCT bit of zero indicates that this drive is not capable of auto-ejecting media.

The NHF (Not HAM Format) bit is used to indicate whether or not the media that is currently installed is properly formatted to the capacity necessary for the HAM format. HAM (Host Adapter Manufacturer) format capacity is the Logical Heads * Logical Cylinders * Logical Sectors * Bytes/sector capacity indicated by the logical format parameters returned in this Mode Page. A NHF bit of zero indicates that the format of the media that is currently inserted supports the HAM format. An NHF bit of one indicates that the format of the media that is currently inserted will not support the HAM format. Usually this would indicate that the defects detected during format are such that the total capacity available on this diskette is below that needed to support the HAM format.

4/11

IMPLEMENTORS NOTE: It is up to the Host Adapter Manufacturer to decide how they wish to treat a diskette that can not support the HAM format. It may be treated as a bad diskette or a diskette with reduced capacity due to "bad sectors" that the user can "see" (for example: shown in the bad sectors field of the response to a DOS CHKDSK command).

[The above line marks the end of suggested text to follow "Table X: Medium Interchange Disk Page"]

CONCLUSION

With the modifications I have suggested here an MI drive will prevent generation of incompatible media formats by non-MI aware SCSI Host Adapter boards. Only an MI-aware SCSI Host Adapter board can properly initialize ("unlock") an MI drive. An MI-aware board can and shall create formats consistent with those as defined by that drive manufacturer by getting and using a 'standard' set of logical parameters for each media type. In this way it will be possible in the future to provide for interchange of direct-access media from drive to drive regardless of the SCSI Host Adapter to which it is attached. I feel that this is an essential next step in making SCSI more accepted throughout the industry as the only choice for a multi-peripheral, multi-media, multi-OS interface for micro-computers.

As always, I am open to any alternative solution that meets the intended goals. I would like to thank Gary Stevens for his assistance on this revision.

[ASIDE: I am very interested in seeing active participation by representatives of companies that develop removable disk or diskette products that are affected by this proposal. Today your removable media DASD drives do not have media interchange among drives attached to different SCSI Host Adapters. This proposal attempts to remedy that in the future. The other suggested solution currently is to define a new device type. Which is preferable for all?]

Attachment A

DESCRIPTION OF MEDIA INTERCHANGE PROBLEM (using DOS references)

Currently, a removable (or fixed) disk drive attached to a SCSI Host Adapter will be partitioned and formatted using a translation structure unique to the manufacturer of the SCSI Host Adapter board and associated BIOS and/or Software. The result is that the removable media may be unreadable if inserted into a similar drive that is attached to a different SCSI Host Adapter (i.e. different manufacturer). It can be worse: the media can appear to be readable since the first part of the disk (Partition table, FAT, Directory) may be in the same place, but the files (made up of clusters) are located differently later on the disk. After use of the disk including some writes, a CHKDSK will show that the disk files have been scrambled. Usually the damage is unrecoverable and the user has just learned a hard lesson at the expense of his disk full of data: "Media is not interchangeable between drives attached to SCSI Host Adapters made by different manufacturers."

WHY TRANSLATION IS NECESSARY

The need to use a translation scheme is due to DOS limitations. However, because of DOS, all other major operating systems (NetWare, Unix, OS/2) also require this logical translation table (the reason differs for each OS). While DOS uses logical blocks (similar to SCSI) it insists on converting to a physical Head, Cylinder, Sector(s) when issuing the INT13 disk I/O call. A SCSI Host Adapter must pick and use some translation scheme to provide for this communication link. Adaptec uses 64 heads, 32 sectors. This keeps the translation the same for all disks on all Adaptec Host Adapters. At the other extreme, some host adapters use 17 sectors. As the disk size increases they will increase heads and sectors as necessary to keep the maximum cylinder number below 1024. There are many reasons for using different translations (for only one example: lower sectors cause older DOS benchmarks to indicate better perceived disk seek times). Other than affecting perceived performance measurements, different translation schemes are neither better or worse ... just different! Today, there is no industry wide interchange of media between identical removable disk drive types attached to SCSI Host Adapters from different makers.

DE FACTO STANDARD ?

I am not sure that there is a De Facto standard. If there were to be a de facto standard ... based on installed base of host adapter products and based on number of manufacturers that use the same scheme ... it would be the translation scheme used by Adaptec of 64 heads and 32 sectors. Although probably as good as any for removable partitionable disk drives, this scheme is not necessarily desirable for all devices such as the new high-capacity SCSI floppy devices. The drive manufacturer should be able to tell every SHA what translation table should be used so that his media has 100% interchange. That is much of what this proposal is about.

[ASIDE: Some manufacturers have different translation schemes among their own products that may vary from disk to disk as capacity increases. Adaptec currently uses the same 64/32 translation scheme across it's entire product line allowing access to any size disk in any OS (except DOS where there is a 1 GByte capacity limit). New Adaptec products allow the user to select an alternate new scheme to allow access to disks over 1 GByte capacity under DOS. This new scheme is not required to access disks over 1 GByte in Unix, NetWare, or OS/2.]

THE FUTURE

Well, who knows what the future holds for sure?? It is a fairly safe guess that some future version of DOS will eliminate the need to convert to this logical translation table for disk access using SCSI. The future DOS logical block may essentially be the same as the SCSI logical block and life will be simpler and more efficient. But, even then, a solution like that proposed in this paper will be necessary to support all back versions of DOS. I do not see the need for an MI solution including the new proposed Mode Page 06h going away for quite some time.