X3T9.2/89-151

Bruce A. Fairman - 275 Martinez Road - Woodside, California 94062
415 851 4156 - MCI BFAIRMAN - AppleLink D1835

27 November 1989

I. Dal Allan
ENDL
14426 Black Walnut Court
Saratoga, CA 95070

Dal :

While working on the Tandon controller, I've developed a (I hope) workable way to use Logical Block format to handle defects in Format. Please let me know if you think this approach has general applicability in the SCSI world.

Regards,

I'd sign this but it's straight form the computer.

Bruce A. Fairman

Bruce A. Fairman - 275 Martinez Road - Woodside, California 94062
415 851 4056 - MCI BFAIRMAN - AppleLink D1835

Format, Read Defects and the Meaning of Defects

B. Fairman 31Oct89
27Nov89

The format of the defect list used by the Format command is (and should be) Logical Block. This defect list format has the advantage of being natural for the Initiator: the Initiator deals only with Logical Blocks. Format will interpret the context of the defects depending on the Format options specified. The goal is to provide a sensible way to scan media and add to the defect list, and also to make manipulation of the defect information possible when special purpose or diagnostic functions are executed.

The actual media flaws are typically recorded in a physical representation: C/H/S, bytes from index, etc. Further, any information about defects exists in one of two spaces: Mapped sector space or Unmapped sector space, depending on the context of the defect information.

Some definitions:

Unmapped sector space: the space defined by all of the sectors which can exist on the media. This is a consistent space, unvarying in the mapping of sector IDs to physical locations. It does not account for media flaws.

Mapped sector space: The "useable" sectors, defined by "mapping" sector IDs around media defects. This is an inconsistent space, varying in the mapping of sector IDs to physical sectors. It accounts for media defects by mapping media defects out of the Logical Block Space.

Logical Blocks: a linear address space of blocks, enumerated from 0 to the maximum number of blocks-1.

Physical Address: the Cylinder, Head and sector location in Mapped sector space.

Mapping: the function M_X, such that M_X(Logical Block) = Physical Address. Where x is the defect list used to construct the mapping.

The PLIST and GLIST make use of the Unmapped sector space, since this in an invariant representation of media flaws. When a target device is scanned for errors (defects), the Initiator operates in Logical Blocks which exist in, for a particular format instance, Mapped sector space. Thus the Format command which succeeds a scan for defects should use Logical Block defects to request the target device add defects to the GLIST, since the Initiator cannot know about the sector mapping which is in effect during the defect scan. Format command execution, since it has access to the PLIST and GLIST, can properly translate the supplied defect list into the Unmapped sector space at the time it constructs the new GLIST from the existing GLIST and supplied defects. This is a valid approach for the Format option which adds to the GLIST, the most common use of Format with a defect list.

There are other options in Format for handling the supplied defects: use the PLIST and: add to the GLIST or replace the GLIST; do not use the PLIST and: add to the GLIST or replace the GLIST. Also, no defects may be supplied during the data out phase. If one suppresses the PLIST and the GLIST and does not supply any defects at Format time, the device is formatted in the Unmapped sector space, the PLIST is retained on the device and the GLIST is set to NULL. If the GLIST is suppressed and no defect data is supplied, then the existing GLIST is lost and the device is formatted to the Mapped sector space generated by the PLIST. All options which do not add to the GLIST supply defects in the Unmapped sector space.

Summary of PLIST, GLIST and DEFECTS actions by FORMAT

CMPLST	DPRY	DFCT	D SPACE	NEW GLIST-G'	NEW SECTOR SPACE
0	0	0	n/c		as before format
0	0	1	M	G'<-G+D	$M_{P+G'}$
0	1	0	n/c		$M_{G'}$
0	1	1	M	G'<-G+D	$M_{G'}$
1	0	0	n/c	G'<- NULL	M_P
1	0	1	U	G'<-D	$M_{P+G'}$
1	1	0	n/c	G'<- NULL	U
1	1	1	U	G'<-D	$M_{G'}$ (U if $D_{count} = 0$)

CMPLST: 1 => replace GLIST during Format
DPRY: 1 => disable PLIST during Format
DFCT: 1 => GLIST defect information supplied to Format (may be null)
D: Defect list supplied during Format (may have 0 elements)

So, what does this mean for Read Defect Data since the information returned by Read Defect Data are always in the Unmapped sector space? The table above indicates the sector space in which defects exist for purposes of Format. This implies that the Format options which can correctly use Read Defect Data are those which do not add to the GLIST. That is, the information stored in the GLIST and PLIST are in Unmapped sector space and cannot, by definition, represent sectors in Mapped sector space. Thus, if an Initiator were to Read Defect Data for the GLIST and then Format adding to the GLIST using that information, the result would not be as intended.

Utility functions can compute the physical location of defects by reading the defects data and applying the information from the Geometry page of mode sense.