To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 25 May 2004
Subject: 04-164r0 SBC-2 Defect descriptor wording corrections

Revision history
Revision 0 (25 May 2004) First revision, extracted from 04-031r1.

Related documents
sbc2r14 - SCSI Block Commands -2 revision 14
04-031r2 SPC-3 SES-2 SBC-2 Miscellaneous diagnostic page corrections (Rob Elliott, HP)

Overview
The Translate Address diagnostic pages use the defect list formats defined in the FORMAT UNIT command. However, they don't always use them for "defects" - non-defective addresses can be translated.

The definitions of the defect descriptors are unclear, particularly the block descriptors, and were broken even more during an SBC-2 editing session. Although possibly corrected in sbc2r14, the wording still needs careful review.

SCSI-2 wording

FORMAT UNIT:

> Each block format defect descriptor (see table 113) specifies a four-byte defective block address that contains the defect.

READ DEFECT DATA:

> NOTE 1 - NOTE 110 The use of the block format is not recommended. There is no universal model that sensibly defines the meaning of the logical block address of a defect. In the usual case, a defect that has been reassigned no longer has a logical block address.
>
> Defect descriptors returned in the block format are vendor-specific.

SBC-1 wording:

FORMAT UNIT:

> Each block format defect descriptor (see Table 6) specifies a four-byte defective block address that contains the defect. Use of the Block format is vendor-specific.

READ DEFECT DATA:

> NOTE 2 - The use of the block format is not recommended. There is no standard model that defines the meaning of the logical block address of a defect. In the usual case, a defect that has been reassigned no longer has a logical block address.
>
> Defect descriptors returned in the block format are vendor-specific.

Suggested approach for SBC-2

Block descriptors contain either a valid user-accessible LBA (from 0..capacity) or a vendor-specific value above the maximum LBA to represent blocks that are mapped out of user-accessible space.

Specific changes include:

a) Rename "defect descriptor" to "address descriptor"
b) Move the definition of CLIST, DLIST, GLIST, and PLIST from the FORMAT UNIT command into the defect model section 4.8.
c) Move the FORMAT UNIT parameter list definition to its own subsection rather than mix it in with the FORMAT UNIT CDB field definitions

d) Rename FORMAT UNIT's "defect list header" to "parameter list header" since, if an initialization pattern is all that is being used, there is no defect involved
e) Clarified the handling of the 4-byte short block address descriptor format in the Translate Address diagnostic pages (where it sits in an 8 byte field)

Suggested changes

3.1.9 data defect list (DLIST): A list of defects sent by the application client to the device server during a FORMAT UNIT command. See ~~5.3~~4.8.

3.1.19 grown defect list (GLIST): All defects sent by the application client to the device server. See ~~5.3~~4.8.

3.1.23 logical unit certification list (CLIST): Defects detected by the device server during an optional certification process performed during the FORMAT UNIT command. See ~~5.3~~4.8.

3.1.31 primary defect list (PLIST): The list of defects that are considered permanent defects. See ~~5.3~~4.8.

4.8 Medium defects

Any medium has the potential for defects that can cause user data to be lost. Therefore, each logical block may contain additional information that allows the detection of changes to the user data caused by defects in the medium or other phenomena, and may also allow the data to be reconstructed following the detection of such a change (e.g., ECC bytes). Some block devices provide the application client control through use of the mode parameters. Some block devices allow the application client to examine and modify the additional information by using the READ LONG and WRITE LONG commands.

Defects may also be detected and managed during processing of the FORMAT UNIT command (see 5.3). The FORMAT UNIT command defines four sources of defect information: the PLIST, CLIST, DLIST, and GLIST. These defects may be reassigned or avoided during the initialization process so that they do not appear in a logical block.

The algorithm may be controlled by the application client, using options in the FORMAT UNIT command. Four sources of defect location information (i.e., defects) are defined as follows:

a) Primary defect list (PLIST). This is the list of defects, which may be supplied by the original manufacturer of the device or medium, that are considered permanent defects. The PLIST is located outside of the application client-accessible logical block space. The PLIST is accessible by the device server (to reference while formatting), but it is not accessible by the application client except through the READ DEFECT DATA command. Once created, the original PLIST shall not be subject to change;
b) Logical unit certification list (CLIST). This list includes defects detected by the device server during an optional certification process performed during the FORMAT UNIT command. This list shall be added to the GLIST;
c) Data defect list (DLIST). This list of defect descriptors may be supplied by the application client to the device server during the the FORMAT UNIT command. This list shall be added to the GLIST. If the DEFECT LIST LENGTH field in the defect list header is set to zero, there is no DLIST; and
d) Grown defect list (GLIST). The GLIST includes all defects sent by the application client or detected by the device server. The GLIST does not include the PLIST. If the CMPLST bit is set to zero, the GLIST shall include DLISTs provided to the device server during the previous and the current FORMAT UNIT commands. The GLIST shall also include:
    A) defects detected by the format operation during medium certification;
    B) defects previously identified with a REASSIGN BLOCKS command (see 5.19); and
    C) defects previously detected by the device server and automatically reallocated.

The block device may automatically reassign defects if allowed by the Read-Write Error Recovery mode page (see 6.3.4).

Defects may also occur after initialization. The application client issues a REASSIGN BLOCKS command (see 5.19) to request that the specified ~~LOGICAL BLOCK ADDRESS~~logical block address be reassigned to a different part of the medium. This operation may be repeated if a new defect appears at a later time. The total number of defects that may be handled in this manner can be specified in the mode parameters.

Defect management on block devices is vendor-specific. Block devices not using a removable medium may optimize the defect management for capacity or performance or both. Some block devices that use a

removable medium do not support defect management ~~(e.g., some floppy disk devices)~~ or use defect management that does not impede the ability to interchange the medium.

5.3 FORMAT UNIT command

5.3.1 FORMAT UNIT command overview

The FORMAT UNIT command (see table 1) formats the medium into application client addressable logical blocks per the application client defined options. In addition, the medium may be certified and control structures may be created for the management of the medium and defects. The degree that the medium is altered by this command is vendor-specific.

Table 1 — FORMAT UNIT command

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

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

a) defect list(s) to be used;
b) defect locations;
c) that logical unit certification be enabled; and
d) exception handling in the event that defect lists are not accessible.

During the format operation, the device server shall respond to commands as follows:

a) In response to all commands except REQUEST SENSE and INQUIRY, the device server 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 device server shall respond as commanded; and
c) In response to the REQUEST SENSE command, unless an error has occurred, the device server shall return a sense key of NOT READY with the additional sense code set to LOGICAL UNIT NOT READY FORMAT IN PROGRESS, with the sense key specific bytes set for progress indication (see SPC-3). See SPC-3 for a description of deferred error handling that may occur during the format operation.

NOTE 3 - The MODE SELECT parameters, if any, should be set prior to issuing the FORMAT UNIT command.

~~During the processing of the FORMAT UNIT command, the device server may perform a medium defect management algorithm. The algorithm may be controlled by the application client, using optional forms of this command. Four sources of defect location information (i.e., defects) are defined as follows:~~

~~a) Primary defect list (PLIST). This is the list of defects, that may be supplied by the original manufacturer of the device or medium, that are considered permanent defects. The PLIST is located outside of~~

~~the application client accessible logical block space. The PLIST is accessible by the device server (to reference while formatting), but it is not accessible by the application client except through the READ DEFECT DATA command. Once created, the original PLIST shall not be subject to change;~~

~~b) Logical unit certification list (CLIST). This list includes defects detected by the device server during an optional certification process performed during the FORMAT UNIT command. This list shall be added to the GLIST;~~

~~c) Data defect list (DLIST). This list of defect descriptors may be supplied by the application client to the device server during the the FORMAT UNIT command. This list shall be added to the GLIST. If the DEFECT LIST LENGTH field in the defect list header is set to zero, there is no DLIST; and~~

~~d) Grown defect list (GLIST). The GLIST includes all defects sent by the application client or detected by the device server. The GLIST does not include the PLIST. If the CMPLST bit is set to zero, the GLIST shall include DLISTs provided to the device server during the previous and the current FORMAT UNIT commands. The GLIST shall also include:~~

~~A) defects detected by the format operation during medium certification;~~

~~B) defects previously identified with a REASSIGN BLOCKS command (see 5.19); and~~

~~C) defects previously detected by the device server and automatically reallocated.~~

A format protection information (FMTPINFO) bit set to zero specifies that the device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header (see SPC-3). A FMTPINFO bit set to one specifies that the device server shall format the medium to the block length specified in the mode parameter block descriptor of the mode parameter header plus eight (e.g., if the block length is 512, then the formatted block length is 520). Following a successful format, the PROT_EN bit in the long read capacity data (see 5.14) indicates whether protection information (see 4.15) is enabled.

If the FMTPINFO bit is set to zero, the device server shall ignore the reference tag own request (RTO_REQ) bit. If the FMTPINFO bit is set to one and the RTO_REQ bit is set to one, the device server shall enable application client ownership of the LOGICAL BLOCK REFERENCE TAG field in protection information (see 4.15). If the FMTPINFO bit set to one and the RTO_REQ bit is set to zero, the device server shall disable application client ownership of the LOGICAL BLOCK REFERENCE TAG field. Following a successful format, the RTO_EN bit in the long read capacity data (see 5.14) indicates whether application client ownership of the LOGICAL BLOCK REFERENCE TAG field is enabled.

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

~~A LONGLIST bit set to zero specifies that the defect list header follows the short format in table 5. A LONGLIST bit set to one specifies that the defect list header follows the long format in table 6.~~

A LONGLIST bit set to zero specifies that the parameter list contains a short parameter list header as defined in table 5. A LONGLIST bit set to one specifies that the parameter list contains a long parameter list header as defined in table 6.

A format data (FMTDATA) bit set to zero specifies that no parameter ~~data~~list be transferred from the application client data-out buffer. The source of defect information is not specified.

A FMTDATA bit set to one specifies that the FORMAT UNIT parameter list (see ~~table 14~~ 5.3.new) shall be transferred from the application client data-out buffer. ~~The parameter list consists of a defect list header, followed by an optional initialization pattern descriptor, followed by zero or more defect descriptors. Each~~

~~defect descriptor specifies a location on the medium that the device server shall map out of the area accessible by application clients.~~

~~Table 2 — FORMAT UNIT parameter list~~

Byte\Bit	7	6	5	4	3	2	1	0
0 to 3 or 0 to 7	~~Defect list header~~							
~~Initialization pattern descriptor (if any) (see 5.3.4)~~								
~~Defect descriptor(s) (if any)~~								
				~~Defect descriptor 0~~				
				~~...~~				
				~~Defect descriptor x~~				

A complete list (CMPLST) bit set to zero specifies that the defect list sent by the application client shall be used in an addition to the existing list of defects. As a result, the device server shall construct a new GLIST (see 4.8) ~~is constructed~~ that contains:

a) the existing GLIST;
b) the DLIST, if it is sent by the application client; and
c) the CLIST, if certification is enabled (i.e., the device server may add any defects it detects during the format operation).

A CMPLST bit set to one specifies that the defect list sent by the application client is a complete list of defects. Any existing defect list except the PLIST shall be ignored by the device server. As a result, the device server shall construct a new GLIST (see 4.8) ~~is constructed~~ that contains:

a) the DLIST, if it is sent by the application client; and
b) the CLIST, if certification is enabled (i.e., the device server may add any defects it detects during the format operation).

The DEFECT LIST FORMAT field specifies the format of the defect descriptors to be used if the FMTDATA bit is set to one (see table 3).

Table 3 defines the ~~defect descriptor requirements~~defect descriptor usage for the FORMAT UNIT command.

Table 3 — FORMAT UNIT defect descriptor usage~~format and requirements~~

Field in the FORMAT UNIT CDB			DEFECT LIST LENGTH field in the defect list header (see 5.3.new.2)	Type [a]	Comments
FMTDATA	CMPLST	DEFECT LIST FORMAT			
0	0	000b	N/A	M	Vendor-specific defect information
1	0	000b (short block)	Zero	M	See notes [b] and [d]
1	1			M	See notes [b] and [e]
1	0		> 0	O	See notes [c] and [d]
1	1			O	See notes [c] and [e]
1	0	011b (long block)	Zero	O	See notes [b] and [d]
1	1			O	See notes [b] and [e]
1	0		> 0	O	See notes [c] and [d]
1	1			O	See notes [c] and [e]
1	0	100b (bytes from index)	Zero	O	See notes [b] and [d]
1	1			O	See notes [b] and [e]
1	0		> 0	O	See notes [c] and [d]
1	1			O	See notes [c] and [e]
1	0	101b (physical sector)	Zero	O	See notes [b] and [d]
1	1			O	See notes [b] and [e]
1	0		> 0	O	See notes [c] and [d]
1	1			O	See notes [c] and [e]
1	0	110b (vendor specific)	Vendor specific		
1	1				
All others				All remaining codes are reserved.	

Notes:
[a] M = implementation is mandatory. O = implementation is optional.
[b] No DLIST is ~~transferred to the device server~~included in the parameter list.
[c] A DLIST is ~~transferred to the device server~~included in the parameter list. ~~Add~~The device server shall add the DLIST defects to the new GLIST.
[d] ~~Use the existing GLIST as a defect source. Add~~The device server shall add existing GLIST defects to the new GLIST (i.e., use the existing GLIST).
[e] ~~Discard the existing GLIST. Do not add~~The device server shall not add existing GLIST defects to the new GLIST (i.e., discard the existing GLIST).

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

Editor's Note 1: if 110b is specified, is the whole parameter list vendor specific or is just the defect list portion? I.e., is the initialization pattern still standard?

Editor's Note 2: Could this table just be dropped? the bit descriptions seem to cover everything already.

5.3.new FORMAT UNIT parameter list

5.3.new.1 FORMAT UNIT parameter list overview

Table 4 defines the FORMAT UNIT parameter list.

Table 4 — FORMAT UNIT parameter list

Byte\Bit	7	6	5	4	3	2	1	0
0 to 3 or 0 to 7	~~Defect~~Parameter list header							
	Initialization pattern descriptor (if any)							
	Defect descriptor(s) (if any)							

The parameter list header is defined in 5.3.new.2.

The initialization pattern descriptor, if any, is defined in 5.3.new.4.

Each defect descriptor contains an address descriptor (see 5.3.new.3) specifies a location on the medium that the device server shall map out of the area accessible by application clients.

5.3.new.2 ~~Defect~~Parameter list header

The ~~defect~~parameter list headers (see table 5 and table 6) provide several optional format control bits. Device servers that implement these bits provide the application client additional control over the use of the four defect sources, and the formatting operation. If the application client attempts to select any function not implemented by the device server, the device server shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Table 5 shows the short ~~defect~~parameter list header, which is used if the LONGLIST bit is set to zero in the FORMAT UNIT CDB (see 5.3.1).

Table 5 — Short ~~defect~~parameter list header

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor specific
2	(MSB)	DEFECT LIST LENGTH						
3								(LSB)

Table 6 shows the long ~~defect~~parameter list header, which is used if the LONGLIST bit is set to one in the FORMAT UNIT CDB (see 5.3.1).

Table 6 — Long ~~defect~~parameter list header

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	FOV	DPRY	DCRT	STPF	IP	Obsolete	IMMED	Vendor specific
2	Reserved							
3	Reserved							
4	(MSB)	DEFECT LIST LENGTH						
7								(LSB)

A format options valid (FOV) bit set to zero specifies that the device server shall use its default settings for the DPRY, DCRT, STPF, and IP bits. If the FOV bit is set to zero, the application client shall set these bits to zero. If the FOV bit is set to zero and any of the other bits in this paragraph are not set to zero, the device server shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

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

A disable primary (DPRY) bit set to zero specifies that the device server shall not use portions of the medium identified as defective in the PLIST for application client addressable logical blocks. If the device server is not able to locate the PLIST or it is not able to determine whether a PLIST exists, it shall perform the action specified by the STPF bit. A DPRY bit set to one specifies that the device server shall not use the PLIST to identify defective areas of the medium. The PLIST is not deleted.

A disable certification (DCRT) bit set to zero specifies that the device server shall perform a vendor-specific medium certification operation to generate a CLIST. A DCRT bit set to one specifies that the device server shall not perform any vendor-specific medium certification process or format verification operation while processing the FORMAT UNIT command.

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

a) The device server has been requested to use the PLIST (i.e., the DPRY bit is set to zero) or the GLIST (i.e., the CMPLST bit is set to zero) and the device server is not able to locate the list nor determine whether the list exists; or
b) The device server has been requested to use the PLIST (i.e., the DPRY bit is set to zero) or the GLIST (i.e., the CMPLST bit is set to zero), and the device server encounters an error while accessing the defect list.

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

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

NOTE 4 - The use of the FMTDATA bit, the CMPLST bit, and the ~~defect~~parameter list header allow the application client to control the source of the defect lists used by the FORMAT UNIT command. Setting the

DEFECT LIST LENGTH field to zero allows the application client to control the use of PLIST and CLIST without having to specify a DLIST.

An initialization pattern (IP) bit set to zero specifies that an initialization pattern descriptor is not included and that the device server shall use its default initialization pattern. An IP bit set to one specifies that an initialization pattern descriptor (see 5.3.4) is included in the FORMAT UNIT parameter list following the ~~defect~~parameter list header.

An immediate (IMMED) bit set to zero specifies that status shall be returned after the format operation has completed. An IMMED bit value set to one specifies that the device server shall return status after the entire ~~defect~~parameter list has been transferred.

The DEFECT LIST LENGTH field ~~in the defect list header~~ specifies the total length in bytes of the defect descriptors that follow and does not include the initialization pattern descriptor or initialization pattern, if any. The length of the defect descriptors varies with the format of the defect list. The ~~three~~ formats for the defect descriptor(s) ~~field in the defect lists~~ are shown in .

The defect descriptors ~~should~~shall be in ascending order. More than one physical or logical block may be affected by each defect descriptor. ~~A device server may return CHECK CONDITION i~~If the defect descriptors are not in ascending order, the device server shall return CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN PARAMETER LIST.

NOTE 5 - Logical units compliant with previous versions of this standard may allow short block format descriptors to not be in ascending order.

Editor's Note 3: the SBC-1 rule was block descriptors should be ascending but bytes from index and physical sector shall be ascending. This suggests mandating they all be ascending. Otherwise, should the long block format be required to be ascending like the short format or not required like the other formats?

5.3.3 ~~Defect list~~ Address descriptor formats

5.3.3.1 ~~Defect list~~ Address descriptor formats overview

This subclause describes the ~~defect list~~ address descriptor formats used in the FORMAT UNIT command, the READ DEFECT DATA commands (see 5.14 and 5.15), and the Translate Address diagnostic pages (see 6.1.2 and 6.1.3) of the SEND DIAGNOSTIC and RECEIVE DIAGNOSTIC RESULTS commands.

The format type of ~~a defect~~an address descriptor is specified with:

a) the DEFECT LIST FORMAT field in the CDB, for the FORMAT UNIT and READ DEFECT DATA commands;
b) the SUPPLIED FORMAT field, for the Translate Address diagnostic pages; or
c) the TRANSLATE FORMAT field, for the Translate Address diagnostic pages.

Table 7 defines the types of ~~defect~~address descriptors.

Table 7 — ~~Defect~~address descriptor formats

Format type	Description	Reference
000b	Short block format ~~defect~~address descriptor	5.3.3.2
011b	Long block format ~~defect~~address descriptor	5.3.3.3
100b	Bytes from index format ~~defect~~address descriptor	5.3.3.4
101b	Physical sector format ~~defect~~address descriptor	5.3.3.5
110b	Vendor-specific	
All others	Reserved	

5.3.3.2 Short block format ~~defect~~address descriptor

A format type of 000b specifies the short block format ~~defect~~address descriptor defined in table 8.

Table 8 — Short block format ~~defect~~address descriptor (000b)

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

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

5.3.3.3 Long block format ~~defect~~address descriptor

A format type of 011b specifies the long block format ~~defect~~address descriptor defined in table 9.

Table 9 — Long block format ~~defect~~address descriptor (011b)

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

For the FORMAT UNIT command, the ~~DEFECTIVE~~ LONG BLOCK ADDRESS field contains the eight-byte logical block address of a defect. For the READ DEFECT DATA command, the ~~DEFECTIVE~~ SHORT BLOCK ADDRESS field contains a vendor-specific eight-byte value. For the Translate Address diagnostic pages, the ~~DEFECTIVE~~ SHORT BLOCK ADDRESS field contains a four-byte LBA or a vendor-specific four byte value above the capacity of the medium.

5.3.3.4 Bytes from index format ~~defect~~address descriptor

A format type of 100b specifies the bytes from index ~~defect~~address descriptor defined in table 10. For the FORMAT UNIT command and READ DEFECT DATA command, this ~~This~~ descriptor specifies the location of a defect that is either the length of one track or no more than eight bytes long. For the Translate Address diagnostic pages, this descriptor specifies the location of a track or the first byte or last byte of an area.

Table 10 — Bytes from index format ~~defect~~address descriptor (100b)

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

The CYLINDER NUMBER ~~OF DEFECT~~ field contains the cylinder number ~~of the defect~~.

The HEAD NUMBER ~~OF DEFECT~~ field contains the head number ~~of the defect~~.

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

~~The defect descriptors shall be in ascending order. The~~ For sorting bytes from index format address descriptors into ascending order, the cylinder number ~~of the defect~~ is the most significant part of the address and the ~~defect~~ bytes from index is the least significant part of the address. More than one logical block may be ~~affected by each defect~~described by this descriptor. ~~A DEFECT BYTES FROM INDEX field set to FFFFFFFFh specifies that the entire track shall be considered defective.~~

5.3.3.5 Physical sector format ~~defect~~address descriptor

A format type of 101b specifies the physical sector ~~defect~~address descriptor defined in table 11. For the FORMAT UNIT command and READ DEFECT DATA command, this ~~This~~ descriptor specifies the location of a defect that is either the length of one track or the length of a sector. For the Translate Address diagnostic pages, this descriptor specifies the location of a track or sector.

Table 11 — Physical sector format ~~defect~~address descriptor (101b)

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)							
	CYLINDER NUMBER ~~OF DEFECT~~							
2								(LSB)
3	HEAD NUMBER ~~OF DEFECT~~							
4	(MSB)							
	~~DEFECT~~ SECTOR NUMBER							
7								(LSB)

The CYLINDER NUMBER ~~OF DEFECT~~ field contains the cylinder number ~~of the defect~~.

The HEAD NUMBER ~~OF DEFECT~~ field contains the head number ~~of the defect~~.

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

~~The defect descriptors shall be in ascending order. The~~ For sorting physical sector format descriptors into ascending order, the cylinder number ~~of the defect~~ is the most significant part of the address and the ~~defect's~~ sector number is the least significant part of the address. More than one logical block may be ~~affected by each defect descriptor~~described by this descriptor. ~~A DEFECT SECTOR NUMBER field set to FFFFFFFFh specifies that the entire track shall be considered defective.~~

5.3.new.4 Initialization pattern descriptor

The initialization pattern descriptor specifies that the logical blocks contain the specified initialization pattern. The initialization pattern descriptor (see table 23) is sent to the device server as part of the FORMAT UNIT parameter list.

The IP MODIFIER field specifies the type and location of a header that modifies the initialization pattern (see table 12).

Table 12 — Initialization pattern modifier

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

Editor's Note 4: moved the 3 SI paragraphs ahead of the INITIALIZATION PATTERN TYPE paragraph

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

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

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

The INITIALIZATION PATTERN TYPE field (see table 13) specifies the type of pattern the device server shall use to initialize each logical block within the application client accessible portion of the medium. All bytes within a

logical block shall be written with the initialization pattern. The initialization pattern is modified by the IP MODIFIER field as described in table 13.

Table 13 — Initialization pattern type

Initialization pattern type	Description
00h	Use default pattern [a]
01h	Repeat the initialization pattern as required to fill the logical block [b]
02h - 7Fh	Reserved
80h - FFh	Vendor-specific

Notes:
[a] If the initialization pattern length is not zero the device server shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.
[b] If the initialization pattern length is zero the device server shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST.

The INITIALIZATION PATTERN LENGTH field specifies the number of bytes contained in the initialization pattern. If the length exceeds the current block length the device server shall terminate the command with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST. The pattern is modified by the IP MODIFIER field.

The INITIALIZATION PATTERN field specifies the initialization pattern.

5.15 READ DEFECT DATA (10) command

...

The DEFECT LIST FORMAT field indicates the format of the defect descriptors returned by the device server. This field is defined in the FORMAT UNIT command (see 5.3.3).

Short block format defect descriptors and long block format defect descriptors returned with this command are vendor-specific. Physical sector format defect descriptors may or may not include defects in areas not accessible to the application client. Bytes from index format defect descriptors shall comprise a complete list of the defects. A complete list of the defects may include defects in areas not within the capacity returned in the READ CAPACITY command.

NOTE 6 - The use of the short block format or the long block format is not recommended for this command. There is no standard model that defines the meaning of the block address of a defect. In the usual case, a defect that has been reassigned no longer has an LBA.

The DEFECT LIST LENGTH field indicates the length in bytes of the defect descriptors that follow. The DEFECT LIST LENGTH is equal to four or eight times the number of the defect descriptors, depending on the format of the returned defect descriptors (see 5.3.3).

If the number of defect descriptors the SCSI device has assigned does not exceed the capability of the ALLOCATION LENGTH field size but contains a value that is insufficient to transfer all of the defect descriptors the defect list length shall not be adjusted to reflect the truncation and the device server shall not create a CHECK CONDITION status. The application client is responsible for comparing the defect list length and the allocation length to determine that a partial list was received. If the number of defect descriptors the SCSI device has assigned exceeds the capability of the ALLOCATION LENGTH field size the device server shall transfer no data and return a CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST and theadditional sense code set to INVALID FIELD IN CDB.

NOTE 7 - The application client may determine the length of the defect list by sending the READ DEFECT DATA (10) command with an ALLOCATION LENGTH of four. The device server returns the defect list header that contains the length of the defect list.

The defect descriptors may or may not be sent in ascending order. The application client may determine the exact number of the defects by dividing the DEFECT LIST LENGTH by the length of a single defect descriptor for the returned format.

5.19 REASSIGN BLOCKS command

The REASSIGN BLOCKS command (see table 48) requests the device server to reassign the defective logical blocks to another area on the medium set aside for this purpose. The device server should also record the location of the defective logical blocks in the GLIST if such a list is supported. More than one physical or logical block may be relocated by each defect descriptor sent by the application client. This command shall not alter the contents of the PLIST (see ~~5.3~~4.8).

6.1 Diagnostic parameters

6.1.2 Translate Address Output diagnostic page

The Translate Address diagnostic pages allow the application client to translate an address in one of the ~~forms~~formats supported by the FORMAT UNIT command (see 5.3) - ~~a~~ short block format address, long block format address, a physical sector format address, or a ~~physical~~ bytes from index format address - into any one of the other formats. The address to be translated is passed to the device server with the SEND DIAGNOSTIC command and the results are returned to the application client by the RECEIVE DIAGNOSTIC RESULTS command. The format of the Translate Address Output diagnostic page sent with SEND DIAGNOSTIC is shown in table 14. The translated address is returned in the Translate Address Input diagnostic page (see table 15).

Table 14 — Translate Address Output diagnostic page

Byte\Bit	7	6	5	4	3	2	1	0
0	PAGE CODE (40h)							
1	Reserved							
2	(MSB)	PAGE LENGTH (000Ah)						
3								(LSB)
4	Reserved					SUPPLIED FORMAT		
5	Reserved					TRANSLATE FORMAT		
6	(MSB)	ADDRESS TO TRANSLATE						
13								(LSB)

The SUPPLIED FORMAT field specifies the format of ADDRESS TO TRANSLATE field. Valid values for this field are defined in the DEFECT LIST FORMAT field of the FORMAT UNIT command (see 5.3). If the device server does not support the requested format it shall terminate the SEND DIAGNOSTIC command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

The TRANSLATE FORMAT field specifies the format the device server shall use for the result of the address translation. Valid values for this field are defined in the DEFECT LIST FORMAT field of the FORMAT UNIT command. If the device server does not support the specified format it shall terminate the command with CHECK CONDITION status, with a sense key set to ILLEGAL REQUEST and an additional sense code set to INVALID FIELD IN PARAMETER LIST.

The ADDRESS TO TRANSLATE field contains a single address the application client is requesting the device server to translate. The format of this field depends on the value in the SUPPLIED FORMAT field. The formats are described in 5.3.3. If the short block format ~~defect~~address descriptor is specified, ~~the block address shall be in~~

~~the first four bytes of the field with the remaining bytes set to zero for four byte addesses and in the ADDRESS TO TRANSLATE field for eight byte addresses..~~ the first four bytes of the ADDRESS TO TRANSLATE field shall conain the short block format address descriptor and the last four bytes shall contain 00000000h.

6.1.3 Translate Address Input diagnostic page

Table 15 defines the Translate Address Input diagnostic page retrieved with RECEIVE DIAGNOSTIC RESULTS after the Translate Address Output diagnostic page has been sent with SEND DIAGNOSTIC. If a Translate Address Output diagnostic page has not yet been processed, the results of a RECEIVE DIAGNOSTIC RESULTS command requesting this diagnostic page are vendor-specific.

Table 15 — Translate Address Input diagnostic page

Byte\Bit	7	6	5	4	3	2	1	0
0	PAGE CODE (40h)							
1	Reserved							
2	(MSB)							
3	PAGE LENGTH (n - 3)							(LSB)
4	Reserved					SUPPLIED FORMAT		
5	RAREA	ALTSEC	ALTTRK	Reserved		TRANSLATED FORMAT		
Translated address(es)								
6	(MSB)							
13	TRANSLATED ADDRESS 1							(LSB)
	…							
n - 7	(MSB)							
n	TRANSLATED ADDRESS x (if required)							(LSB)

The Translate Address Input diagnostic page contains a four-byte page header that specifies the page code and length followed by two bytes that describe the translated address followed by zero or more translated address(s).

The PAGE LENGTH field contains the number of parameter bytes that follow.

The SUPPLIED FORMAT field contains the value from the ~~SEND DIAGNOSTIC command~~ SUPPLIED FORMAT field in the previous Translate Address Output diagnostic page (see 6.1.2).

A reserved area (RAREA) bit set to zero indicates that no part of the translated address falls within a reserved area of the medium. A RAREA bit set to one indicates that all or part of the translated address falls within a reserved area of the medium (e.g., speed tolerance gap, alternate sector, vendor reserved area, etc.). If the entire translated address falls within a reserved area, the device server may not return a translated address.

An alternate sector (ALTSEC) bit set to zero indicates that no part of the translated address is located in an alternate sector of the medium or that the device server is unable to determine this information. An ALTSEC bit set to one indicates that the translated address is physically located in an alternate sector of the medium. If the device server is unable to determine if all or part of the translated address is located in an alternate sector it shall set this bit to zero.

An alternate track (ALTTRK) bit set to zero indicates that no part of the translated address is located on an alternate track of the medium. An ALTTRK bit set to one indicates that part or all of the translated address is located on an alternate track of the medium or the device server is unable to determine if all or part of the translated address is located on an alternate track.

The TRANSLATED FORMAT field contains the value from the ~~Translate Address Output diagnostic page's~~ TRANSLATE FORMAT field in the previous Translate Address Output diagnostic page (see 6.1.2).

The TRANSLATED ADDRESS field(s) contains the address(es) the device server translated from the address supplied by the application client in the ~~SEND DIAGNOSTIC command~~ previous Translate Address Output diagnostic page. ~~This~~ Each field shall be in the format specified in the TRANSLATE FORMAT field. The different formats are described in 5.3.3. If the short block format ~~defect~~address descriptor is specified, ~~the block address shall be in the first four bytes of the field with the remaining bytes set to zero for four byte addesses and in the ADDRESS TO TRANSLATE field for eight byte addresses..~~ the first four bytes of the ADDRESS TO TRANSLATE field shall conain the short block format address descriptor and the last four bytes shall contain 00000000h.

If the returned data is in the ~~logical block~~ short block format, long block format, or physical sector format and the ~~address to be translated~~ ADDRESS TO TRANSLATE field in the previous Translate Address Output diagnostic page covers more than one address after it has been translated (e.g., accounting for ~~speed tolerance or~~ multiple physical sectors within a single logical block or multiple logical blocks within a single physical sector) the device server shall return all possible addresses that are contained in the area specified by the address to be translated.

If the returned data is in bytes from index format, the device server shall return a pair of translated values for each of the possible addresses that are contained in the area specified by the ADDRESS TO TRANSLATE field in the previous Translate Address Output diagnostic page. Of the pair of translated values returned, the first indicates the starting location and the second the ending location of the area.