T10/05-374 revision 2

Date: November 10, 2005

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: SBC-3: SPC-4: Disabling Reassign on Write Long Logical Blocks

1 Overview

Some SCSI/SAS hosts controllers make use of the Write Long command to intentionally create unrecoverable errors on the media. Currently there is no way to differentiate these pseudo uncorrectable errors (intentionally created) with a typical unrecoverable error caused by media defects. There is also a problem as these errors are included in the algorithm used for Information Exception Conditions and, as a result, may cause information exception condition trips, as well as unnecessary reassignments.

The solution being proposed is to add a bit to the Write Long command that adds a bit to the WRITE LONG command that would indicate to a logical unit to not count the logical block being written in any IEC or automatic reassignment calculations. Also, a bit is proposed to be added to the Extended INQUIRY data VPD page to indicate if a logical unit supports this new feature.

2 SBC-3 changes

2.1 Medium defects

Any medium has the potential for defects that cause data to be lost. Therefore, each logical block may contain additional information that allows the detection of changes to the user data and protection information, if any, 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~~ Direct-access block devices may allow the application client to examine and modify the additional information by using the READ LONG commands and the WRITE LONG commands (see 5.14, 5.15, 5.33, and 5.34). The application client may use the WRITE LONG commands to induce a defect to test the defect detection logic of the direct-access block device or to emulate an unrecoverable logical block when generating a mirror copy.

Direct-access block devices may allow the application client to disable error correction and automatic reallocation on specific logical blocks by using the WRITE LONG command (see 2.2). This allows an application client to prevent logical blocks from being included in the algorithm used for information exception conditions there by preventing unwarranted information exception condition trips and unnecessary reassignments

Defects may also be detected and managed during processing of the FORMAT UNIT command (see 5.2). 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 affect any logical blocks. The 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 for reference during the format operation, but it is not accessible by the application client except through the READ DEFECT DATA commands (see 5.10 and 5.13). Once created, the original PLIST shall not 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 defects may be supplied by the application client to the device server during the FORMAT UNIT command. This list shall be added to the GLIST; and
d) Grown defect list (GLIST). The GLIST includes all defects sent by the application client (i.e., the DLIST) or detected by the device server (i.e., the CLIST). 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 5.16); and
C) defects previously detected by the device server and automatically reallocated.

The direct-access block device may automatically reassign defects if allowed by the Read-Write Error Recovery mode page (see 6.3.5).

Defects may also occur after initialization. The application client issues a REASSIGN BLOCKS command (see 5.16) to request that the specified 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 is vendor-specific.

Defect management on direct-access block devices is vendor-specific. Direct-access block devices not using a removable medium may optimize the defect management for capacity or performance or both. Some direct-access block devices that use a removable medium do not support defect management or use defect management that does not impede the ability to interchange the medium.

2.2 WRITE LONG (10) command

The WRITE LONG (10) command (see table 1) requests that the device server transfer data for a single logical block from the data-out buffer and write it to the medium. The data written shall be the same length and shall be in the same order as the data returned by the READ LONG (10) command (see 5.14). The device server shall write the logical block to the medium, and shall not return GOOD status until the logical block has actually been written on the medium.

Table 1 — WRITE LONG (10) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Fh)							
1	COR_DIS	Reserved						Obsolete
2	(MSB)	LOGICAL BLOCK ADDRESS						
5								(LSB)
6	Reserved							
7	(MSB)	BYTE TRANSFER LENGTH						
8								(LSB)
9	CONTROL							

See the PRE-FETCH (10) command (see 5.3) for the definition of the LOGICAL BLOCK ADDRESS field.

A correction disabled (COR_DIS) bit set to zero specifies that, when the specified logical block is read, the device server shall perform normal error recovery on that logical block. A COR_DIS bit set to one specifies specifies that, when the specified logical block is read, the device server shall:

a) perform no error recovery on that logical block including any read error recovery enabled by the Read-Write Error Recovery mode page (see 2.3.1);
b) perform no automatic reallocation of that logical block including any automatic reallocation enabled by the Read-Write Error Recovery mode page;
c) not consider errors on logical blocks to be informational exception conditions as defined in the Information Exceptions Control mode page (see SPC-4): and
d) return CHECK CONDITION status with the sense key set to MEDIUM ERROR and the additional sense code set to READ ERROR - LBA MARKED BAD BY APPLICATION CLIENT.

The condition established by the COR_DIS bit being set to one shall remain in effect until the logical block is written by any means (e.g., any WRITE command, WRITE SAME command, FORMAT command, or another WRITE LONG command specifying the same logical block with the COR_DIS bit set to zero).

The BYTE TRANSFER LENGTH field specifies the number of bytes of data that the device server shall transfer from the data-out buffer and write to the specified logical block. If the BYTE TRANSFER LENGTH field is not set to zero and does not match the data length that the device server returns for a READ LONG command, then the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB. In the sense data (see 4.13 and SPC-4), the ILI and VALID bits shall be set to one and the INFORMATION field shall be set to the difference (i.e., residue) of the requested length minus the actual length in bytes. Negative values shall be indicated by two's complement notation. A BYTE TRANSFER LENGTH field set to zero specifies that no bytes shall be written. This condition shall not be considered an error.

2.3 WRITE LONG (16) command

The WRITE LONG (16) command (see table 2) requests that the device server transfer data for a single logical block from the data-out buffer and write it to the medium. The data written shall be the same length and shall be in the same order as the data returned by the READ LONG (16) command (see 5.15). The device server shall write the logical block to the medium, and shall not return GOOD status until the logical block has actually been written on the medium. This command is implemented as a service action of the SERVICE ACTION OUT operation code (see A.2).

Table 2 — WRITE LONG (16) command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (9Fh)							
1	COR_DIS	Reserved		SERVICE ACTION (11h)				
2	(MSB)	LOGICAL BLOCK ADDRESS						
9								(LSB)
10		Reserved						
11								
12	(MSB)	BYTE TRANSFER LENGTH						
13								(LSB)
14	Reserved							
15	CONTROL							

See the WRITE LONG (10) command (see 2.2) for the definitions of the fields in this command.

2.3.1 Read-Write Error Recovery mode page

The Read-Write Error Recovery mode page (see table 3) specifies the error recovery parameters the device server shall use during any command that performs a read or write operation to the medium (e.g., READ

commands, WRITE commands, and WRITE AND VERIFY commands), except on logical blocks that have correction disabled (see 2.2).

Table 3 — Read-Write Error Recovery mode page

Byte\Bit	7	6	5	4	3	2	1	0
0	PS	Reserved	PAGE CODE (01h)					
1	PAGE LENGTH (0Ah)							
2	AWRE	ARRE	TB	RC	Error recovery bits			
					EER	PER	DTE	DCR
3	READ RETRY COUNT							
4	Obsolete							
5	Obsolete							
6	Obsolete							
7	Reserved						Restricted for MMC-4	
8	WRITE RETRY COUNT							
9	Reserved							
10	(MSB)	RECOVERY TIME LIMIT						
11								(LSB)

....

3 SPC-4 changes

3.1 New ASC/ASCQ

11h 14h READ ERROR - LBA MARKED BAD BY APPLICATION CLIENT

3.1.1 Extended INQUIRY Data VPD page

The Extended INQUIRY Data VPD page (see table 4) provides the application client with a means to obtain information about the logical unit.

Table 4 — Extended INQUIRY Data VPD page

Bit Byte	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (86h)							
2	Reserved							
3	PAGE LENGTH (3Ch)							
4	Reserved				RTO	GRD_CHK	APP_CHK	REF_CHK
5	Reserved			GROUP_SUP	PRIOR_SUP	HEADSUP	ORDSUP	SIMPSUP
6	Reserved					COR_D_SUP	NV_SUP	V_SUP
7	Reserved							
63								

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are as defined in 6.4.2.

The PAGE LENGTH field specifies the length of the following VPD page data and shall be set to 60. The relationship between the PAGE LENGTH field and the CDB ALLOCATION LENGTH field is defined in 4.3.4.6.

....

A correction disable supported (COR_D_SUP) bit set to zero indicates that the device server does not support application clients disabling read error checking on a logical block written using a WRITE LONG command (see SBC-3). A COR_D_SUP bit set to one indicates that the device server supports application client disabling read error checking on a logical block written using a WRITE LONG command.