

Date: December 05, 2003
 To: T10 Committee (SCSI)
 From: George Penokie (IBM/Tivoli)
 Subject: SBC-2: Write Same fixes

1 Overview

When the WRITE SAME (16) command was defined to handle the 8 byte LBAs there was no statement made about what to do if LBDATA bit is set to one. The current description only allows 4 bytes of LBA to be written. That is an error that is corrected in this proposal by requiring the full 8 byte of LBA to be written when the WRITE SAME (16) is used with the LBDATA bit set to one.

The other problem with the WRITE SAME command is the when protection information is enabled the and setting the LBDATA bit or the PBDATA bit to one will cause the CRC to be regenerated on each logical block. This proposal allows the target to reject the WRITE SAME command as an Illegal Request if either of those bits are set.

2 SBC-2 changes

2.0.1 WRITE SAME (10) command

The WRITE SAME (10) command (see table 1) requests that the device server write the single block of data transferred from the application client to the medium multiple times to consecutive multiple logical blocks.

NOTE 1 - This command may be useful if large areas of the medium need to be written, prepared for certification, or otherwise initialized without the application client having to transfer all the data.

Table 1 — WRITE SAME (10) command

Byte/Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (41h)							
1	WRPROTECT			Reserved		PBDATA	LBDATA	Obsolete
2	(MSB) _____							
5	LOGICAL BLOCK ADDRESS _____ (LSB)							
6	Reserved							
7	(MSB) _____							
8	NUMBER OF BLOCKS _____ (LSB)							
9	CONTROL							

See 4.2.1.9 for reservation requirements for this command. See the LOCK UNLOCK CACHE (10) command (see 5.2.3) for a definition of the LOGICAL BLOCK ADDRESS field.

If the medium is formatted with protection information the value in the DATA BLOCK REFERENCE TAG field received from the application client shall be placed into the DATA BLOCK REFERENCE TAG field (see 4.5.2) of the first logical block written to the medium. Into each of the following logical blocks the data block reference tag received in the data transferred from the application client, incremented by one, shall be placed into the DATA BLOCK REFERENCE TAG field of that logical block (i.e., each logical block written to the medium has a data block reference tag value of one greater than the previous logical block).

If the APP_TAG_OWN bit in the Control mode page (see SPC-3) is set to one, the data block application tag received in the single block of data shall be placed in the DATA BLOCK APPLICATION TAG field of each logical

block. If the APP_TAG_OWN bit is set to zero, the data block application tag received in the single block of data may be placed in the DATA BLOCK APPLICATION TAG field of each logical block.

~~A logical block data (LBDATA) bit of zero and a physical block data (PBDATA) bit of zero indicates that the single block of data transferred from the application client shall be used without modification. A LBDATA bit of one requests that the device server replace the first four bytes of the data to be written to the current logical block with the logical block address of the block currently being written.~~

~~A PBDATA bit of one requests that the device server replace the first eight bytes of the data to be written to the current physical sector with the physical address of the sector currently being written using the physical sector format (see 5.2.2.2).~~

~~If PBDATA and LBDATA are one the command shall be terminated with CHECK CONDITION status and the sense key shall be set to ILLEGAL REQUEST with the appropriate additional sense code for the condition.~~

[\[replace with:\]](#)

Table 2 describes the LBDATA bit and the PBDATA bit.

Table 2 — LBDATA bit and PBDATA bit

LBDATA	PBDATA	Description
0	0	The device server shall write the single block of data received from the application client to each logical block without modification.
0	1 ^a	The device server shall replace the first eight bytes of the block received from the application client to each physical sector with the physical address of the sector being written using the physical sector format (see 5.2.2.2).
1 ^a	0	The device server shall replace the first bytes of the block received from the application client with the LBA of the block being written (i.e., bytes 0-3 of the logical block if the logical block address is four bytes or byte 0-7 of the logical block if the logical block address is eight bytes).
1	1	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.
^a If the medium is formatted with protection information then the logical unit is required to recalculate a CRC for each logical block written to the medium and place the new CRC value into the corresponding logical block guard field. If the logical unit does not support recalculation of the CRC the device server shall terminate the command with a CHECK CONDITION status with a sense key set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN CDB.		

Editor's Note 1: This proposal requires the use of the full LBA regardless of it's size. But in the FORMAT UNIT command the Initialization pattern modifier option that places an LBA in each block only allows four bytes to be used regardless of the actual LBA size. Write Same and Format should work the same way. The question is which way?
