EMC^2®
where information lives®

To: INCITS Technical Committee T10
From: David L. Black, EMC
Email: black_david@emc.com
Date: Sep 4, 2008
Subject: SBC-3: TRIM bit for WRITE SAME (08-356r0)

1) *Revision history*

Revision 0 (Sep 4, 2008) First revision (r0)

2) *Related documents*

spc4r16 – SCSI Primary Commands – 4
sbc3r16 – SCSI Block Commands – 3
ssc3r04a – SCSI Sequential Commands – 3
08-149r1 – Thin Provisioning Commands
08-347r0 – TRIM: Behavior of subsequent READs
T13/e07154r6 – ATA8-ACS2 accepted TRIM proposal

3) *Overview*

08-149r1 proposes a new TRIM command to allow a device server to deallocate physical blocks for logical blocks whose contents are no longer needed. The read behavior of logical blocks affected by the command is not deterministic; the device server chooses what (if any) physical blocks to deallocate and chooses what value to report when a READ command is processed for a logical block whose physical block has been deallocated. The resulting flexibility is very useful to implementations.

This read behavior fits use cases where the value of the data read from a logical block that has been trimmed is not important. A disk or file system defragmenter provides an example of such a use case because the defragmenter does not care what the results of reading the newly freed space are and the filesystem does not rely on the contents of the newly freed space.

There are also use cases in which the value of the data read from a logical block after it has been trimmed is important. Filesystems provide a class of examples because:

- It is useful for filesystems to allow deallocation of the physical blocks for logical blocks that are not going to be used for a significant amount of time.
- When those logical blocks are put back into use (e.g., in response to increased space demand), the blocks have to be initialized, often to zero.

Many filesystems have cases in which the filesystem must initialize blocks being placed into use; these include partial block writes and indirect blocks. If the SCSI operation that allows the device server to deallocate physical blocks also initializes all of the affected logical blocks, those blocks do not have to be re-initialized in order to return them to use. The has the potential to shift initialization operations from high demand times (when space that has been free for some time is needed) to lower demand times when I/O resources are more readily available.

This document proposes adding a TRIM bit to the WRITE SAME commands in order to address this class of use cases exemplified by filesystems.

Existing text is shown in **BLACK**, new text is shown in **RED**, and comments (not to be included) are shown in **BLUE**.

Proposal:

5.37 WRITE SAME (10) command

Add a TRIM bit to the CDB:

Table 88 – WRITE SAME (10) Command

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

The OPERATION CODE field is defined in SPC-4 and shall be set to the value defined in table 88.

See the WRITE (10) command (see 5.27) for the definitions of the CONTROL byte and WRPROTECT field. See the PRE-FETCH (10) command (see 5.4) for the definition of the LOGICAL BLOCK ADDRESS field. See the PRE-FETCH (10) command (see 5.4) and 4.18 for the definition of the GROUP NUMBER field.

If the logical unit is thin provisioned (see 4.4.x), a TRIM bit set to one designates any physical blocks that currently store user data for the specified range of LBAs as candidates for deallocation; the device server may deallocate some or all of the physical blocks. A TRIM bit set to zero has no effect. If the logical unit is not thin provisioned, the device server shall ignore the TRIM bit.

The value of the TRIM bit shall not affect the data obtained by subsequent commands that read or otherwise use the specified logical blocks; the commands shall behave as if the TRIM bit was ignored.

5.38 WRITE SAME (16) command

Add a TRIM bit to the CDB:

Table 90 – WRITE SAME (16) Command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (93h)							
1	WRPROTECT			**Reserved**	**TRIM**	PBDATA	LBDATA	Reserved
2	(MSB)		LOGICAL BLOCK ADDRESS					
9								(LSB)
10	(MSB)		NUMBER OF LOGICAL BLOCKS					
13								(LSB)
14	Reserved			GROUP NUMBER				
15	CONTROL							

The OPERATION CODE field is defined in SPC-4 and shall be set to the value defined in table 90.

See the WRITE SAME (10) command (see 5.37) for the definitions of the other fields in this command. <No change is needed to this text.>

5.38 WRITE SAME (32) command

Add a TRIM bit to the CDB:

Table 91 – WRITE SAME (32) Command

Bit Byte	7	6	5	4	3	2	1	0
0	OPERATION CODE (7Fh)							
1	CONTROL							
2	Reserved							
5								
6	Reserved			GROUP NUMBER				
7	ADDITIONAL CDB LENGTH (18h)							
8	(MSB)		SERVICE ACTION (000Dh)					
9								(LSB)
10	WRPROTECT			**Reserved**	**TRIM**	PBDATA	LBDATA	Obsolete
11	Reserved							
12	**No change to remainder of CDB fields**							
31								

The OPERATION CODE field, ADDITIONAL CDB LENGTH field, and SERVICE ACTION field are defined in SPC-4 and shall be set to the values defined in table 91.

See the WRITE SAME (10) command (see 5.37) for the definitions of the CONTROL byte, GROUP NUMBER field, the WRPROTECT field, the TRIM bit, the PBDATA bit, the LBDATA bit, the LOGICAL BLOCK ADDRESS field, and the NUMBER OF LOGICAL BLOCKS field.