ENDL
TEXAS

Date: 2 July 2007
To: T10 Technical Committee
From: Ralph O. Weber
Subject: OSD-2 CLEAR command, PUNCH command, & range-based FLUSH

Introduction

The SNIA OSD TWG has requested the addition of two new commands (CLEAR and PUNCH) and enhancement to the FLUSH command.

Revision History

r0 Initial revision
r1 Revised to address the comments from Rob Elliott shown below

Differences between r0 and r1 are identified by change bars.

Unless otherwise indicated additions are shown in blue, deletions in ~~red strikethrough~~, and comments in green.

Rob Elliott comments that motivated r1

[r0] Page 3: This is existing wording, so you may choose to ignore this comment. The phrase "shall be written" in table 58 and in lots of the new text is slightly overstated. If the data has already been written to stable storage, then the FLUSH command doesn't actually force another write.

[r0] Page 10: If an object is (somehow) already beyond the quota, and you run a PUNCH command to shorten it to a size that is still beyond the quota? Is that always allowed, or would the quota rules prevent modifying it? It doesn't quite "attempt to consume additional applicable resource" if just the data itself is considered, but the metadata to remember the gap might be considered as increased.

Proposed Changes in OSD-2 r01

18484 Preston Road, Suite 102 PMB 178, Dallas, TX 75252 214-912-1373 972-596-2775 Fx ENDL_TX@computer.org

Change 1 – Range-based FLUSH

6.7 FLUSH

The FLUSH command (see table 57) ensures that the specified data and attribute bytes for the specified user object are written to stable storage (see 4.11).

Table 57 — FLUSH command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8888h)				(LSB)
10				Reserved				FLUSH SCOPE
11	Reserved			GET/SET CDBFMT		Reserved		
12				TIMESTAMPS CONTROL				
13								
15				Reserved				
16	(MSB)							
23				PARTITION_ID				(LSB)
24	(MSB)							
31				USER_OBJECT_ID				(LSB)
32	(MSB)							
39				FLUSH LENGTH				(LSB)
40	(MSB)							
47				FLUSH STARTING BYTE ADDRESS				(LSB)
48								
51				Reserved				
~~32~~								
~~51~~				~~Reserved~~				
52								
79				Get and set attributes parameters (see 5.2.2)				
80								
159				Capability (see 4.9.2.2)				
160								
199				Security parameters (see 5.2.6)				

The FLUSH SCOPE field (see table 58) specifies the scope of the data and attribute bytes that are written to stable storage.

Table 58 — User object flush scope values

Value	Scope of data and attributes that the device server shall ~~be~~ ensure are written to stable storage	Range fields reserved [a]
00b	User object data and attributes	Yes
01b	User object attributes only	Yes
10b	User object data range and attributes	No
~~10b to~~ 11b	Reserved	Yes

[a] The range fields are the FLUSH LENGTH field and the FLUSH STARTING BYTE ADDRESS field.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2.

The contents of the TIMESTAMPS CONTROL field are defined in 5.2.8.

The contents of the PARTITION_ID field are defined in 5.2.5.

The contents of the USER_OBJECT_ID field are defined in 5.2.9.

If the FLUSH SCOPE field contains 10b, the FLUSH LENGTH field specifies number of bytes that the device server shall ensure are written to stable storage.

If the FLUSH SCOPE field contains 10b, the FLUSH STARTING BYTE ADDRESS field specifies the location of the first byte of the flush length bytes that the device server shall ensure are written to stable storage relative to the first byte (i.e., byte zero) of the specified user object.

If the FLUSH SCOPE field contains 10b and the FLUSH STARTING BYTE ADDRESS field specifies a byte that is beyond the user object logical length attribute value in the User Object Information attributes page (see 7.1.2.11), then:

a) No bytes shall be written to stable storage; and
b) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the FLUSH SCOPE field contains 10b, and the values in the FLUSH LENGTH field and FLUSH STARTING BYTE ADDRESS field result an attempt to write a byte that is beyond the user object logical length attribute value in the User Object Information attributes page to stable storage, then the device server shall ensure that the bytes between the flush starting byte address and the user object logical length are written to stable storage. This shall not be considered an error.

If the FLUSH SCOPE field contains 10b, an attempt to flush bytes that have never been written shall result in zeros being written to stable storage for those bytes. This shall not be considered an error.

The command data segment of the Data-Out Buffer is not used by the FLUSH command.

The get and set attributes parameters are defined in 5.2.2. The format of the Data-In Buffer and Data-Out Buffer when attributes are being retrieved or set is described in 4.12.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

Change 2 – Table updates for PUNCH and CLEAR commands

{{Unless otherwise indicated, insertions are in alphabetical order by command name.}}

4.9.2.2 Capability format

4.9.2.2.1 Introduction

…

A READ bit set to one allows read access to the data in an OSD object, but not to the attributes. For the root object, partitions, and collections the data in the OSD object is the list of other objects contained in the OSD object. A READ bit set to zero prohibits read access to the data in an OSD object.

A WRITE bit set to one allows processing of the WRITE command (see 6.28) or an equivalent command, but not access to user object attributes. A WRITE bit set to zero prohibits processing of the WRITE command.

…

Table 18 — Commands allowed by specific capability field values

Commands allowed and CDB fields whose contents are restricted by capability field contents, if any	Capability Field values that allow a command		
	Object Type Name	Permission Bits That Are Set To One	Object Descriptor Name
…	…	…	…
A CLEAR command	USER	WRITE	U/C
…	…	…	…
A PUNCH command	USER	WRITE	U/C
…	…	…	…
Combinations of OBJECT TYPE field, PERMISSION BITS field, and OBJECT DESCRIPTOR TYPE field values not shown in this table and table 19 are reserved. The capability fields not shown in this table may place additional limits on the objects that are allowed to be accessed.			

…

Table 42 — OSD commands that are allowed in the presence of various reservations

OSD Command	Addressed logical unit has this type of persistent reservation held by another I_T nexus				
	From any I_T nexus		From registered I_T nexus (RR all types)	From not registered I_T nexus	
	Write Excl	Excl Access		Write Excl RR	Excl Access – RR
…	…	…	…	…	…
CLEAR	Conflict	Conflict	Allowed	Conflict	Conflict
…	…	…	…	…	…
PUNCH	Conflict	Conflict	Allowed	Conflict	Conflict
…	…	…	…	…	…
Key: **Excl**=Exclusive, **RR**=Registrants Only or All Registrants					

…

Table 51 — Commands for OSD type devices

Command name	Operation code	Service action [a]	Type	Reference
…	…	…	…	…
CLEAR	7Fh	8889h	M	6.x
…	…	…	…	…
PUNCH	7Fh	8884h	M	6.y
…	…	…	…	…

Type Key: M = Command implementation is mandatory.
O = Command implementation is optional.

[a] No entry in the service action column means that the SERVICE ACTION field does not apply to the command. Service action codes values between 8800h and 8F7Fh that are not listed in this table are reserved for future standardization. Service action code values between 8F80h and 8FFFh may have vendor specific command assignments.

[b] …

…

Table B.1 — Numerical order OSD service action codes

Service Action	Command
…	…
8881h	FORMAT OSD
8882h	CREATE
8883h	LIST
8884h	~~Reserved~~ PUNCH
8885h	READ
8886h	WRITE
8887h	APPEND
8888h	FLUSH
8889h	~~Reserved~~ CLEAR
888Ah	REMOVE
888Bh	CREATE PARTITION
888Ch	REMOVE PARTITION
888Dh	Reserved
888Eh	GET ATTRIBUTES
888Fh	SET ATTRIBUTES
…	…

Change 3 – CLEAR and PUNCH commands definitions

{{All text in change 3 is new. Changes markups are suspended for all of change 3.}}

6.x CLEAR

The CLEAR command (see table x1) causes the specified number of bytes containing zero to be written to the specified user object at the specified relative location.

Table x1 — CLEAR command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8889h)				(LSB)
10	Reserved							
11	Reserved		GET/SET CDBFMT		Reserved			
12	TIMESTAMPS CONTROL							
13				Reserved				
15								
16	(MSB)							
23				PARTITION_ID				(LSB)
24	(MSB)							
31				USER_OBJECT_ID				(LSB)
32	(MSB)							
39				CLEAR LENGTH				(LSB)
40	(MSB)							
47				CLEAR STARTING BYTE ADDRESS				(LSB)
48				Reserved				
51								
52				Get and set attributes parameters (see 5.2.2)				
79								
80				Capability (see 4.9.2.2)				
159								
160				Security parameters (see 5.2.6)				
199								

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2.

The contents of the TIMESTAMPS CONTROL field are defined in 5.2.8.

The contents of the PARTITION_ID field are defined in 5.2.5.

The contents of the USER_OBJECT_ID field are defined in 5.2.9.

The CLEAR LENGTH field specifies the number of bytes containing zero to be written to the specified user object.

The CLEAR STARTING BYTE ADDRESS field specifies the location where the writing of bytes containing zero is to commence relative to the first byte (i.e., byte zero) of the specified user object.

Writing zero to a byte at a location that is greater than or equal to the value in the user object logical length attribute in the User Object Information attributes page (see 7.1.2.11) shall implicitly increase the value in the user object logical length attribute to the largest location of any byte written.

The command data segment of the Data-Out Buffer is not used by the CLEAR command.

The get and set attributes parameters are defined in 5.2.2. The format of the Data-In Buffer and Data-Out Buffer when attributes are being retrieved or set is described in 4.12.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

If a CLEAR command causes the value in the user object logical length attribute in the User Object Information attributes page (see 7.1.2.11) to exceed the value in the maximum user object length attribute in the User Object Quotas attributes page, then a quota error shall be generated (see 4.8.2). The quota testing principles described in 4.8.3 apply to the testing of the maximum user object length quota.

If a CLEAR command causes the value in the used capacity attribute in the Partition Information attributes page (see 7.1.2.9) to exceed the value in the capacity quota attribute in the Partition Quotas attributes page (see 7.1.2.13), then a quota error shall be generated. The quota testing principles described in 4.8.3 apply to the testing of the capacity quota.

6.y PUNCH

The PUNCH command (see table x2) removes bytes from a user object.

Table x2 — PUNCH command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8884h)				(LSB)
10	Reserved							
11	Reserved		GET/SET CDBFMT		Reserved			
12	TIMESTAMPS CONTROL							
13				Reserved				
15								
16	(MSB)							
23				PARTITION_ID				(LSB)
24	(MSB)							
31				USER_OBJECT_ID				(LSB)
32	(MSB)							
39				PUNCH LENGTH				(LSB)
40	(MSB)							
47				PUNCH STARTING BYTE ADDRESS				(LSB)
48				Reserved				
51								
52				Get and set attributes parameters (see 5.2.2)				
79								
80				Capability (see 4.9.2.2)				
159								
160				Security parameters (see 5.2.6)				
199								

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2.

The contents of the TIMESTAMPS CONTROL field are defined in 5.2.8.

The contents of the PARTITION_ID field are defined in 5.2.5.

The contents of the USER_OBJECT_ID field are defined in 5.2.9.

The PUNCH LENGTH field specifies number of bytes to be removed from the user object. A punch length of zero shall cause no bytes to be removed from the user object. This shall not be considered an error.

The PUNCH STARTING BYTE ADDRESS field specifies the location where the removal of bytes from the specified user object is to commence relative to the first byte (i.e., byte zero) of the user object (e.g., if the punch starting byte address is five and the punch length is two, then byte seven in the user object becomes byte five, and so on for the remaining logical length of the user object).

If the PUNCH STARTING BYTE ADDRESS field specifies a byte that is beyond the user object logical length attribute value in the User Object Information attributes page (see 7.1.2.11), then:

a) No bytes shall be removed from the user object; and
b) The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the values in the PUNCH LENGTH field and PUNCH STARTING BYTE ADDRESS field result an attempt to remove bytes that are beyond the user object logical length attribute value in the User Object Information attributes page, then the user object shall be truncated by setting the user object logical length attribute value in the User Object Information attributes page to the value in the PUNCH STARTING BYTE ADDRESS field. This shall not be considered an error.

The command data segment of the Data-Out Buffer is not used by the PUNCH command.

The get and set attributes parameters are defined in 5.2.2. The format of the Data-In Buffer and Data-Out Buffer when attributes are being retrieved or set is described in 4.12.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

If a PUNCH command causes the value in the used capacity attribute in the Partition Information attributes page (see 7.1.2.9) to exceed the value in the capacity quota attribute in the Partition Quotas attributes page (see 7.1.2.13), then a quota error shall be generated (see 4.8.2). The quota testing principles described in 4.8.3 apply to the testing of the capacity quota.