

ENDL TEXAS

Date: 28 January 2008
 To: SNIA OSD TWG and T10 Technical Committee
 From: Ralph O. Weber
 Subject: OSD-2 Scatter/Gather features

Introduction

Pete Wyckoff (Ohio Supercomputing Center) has proposed adding scatter/gather capabilities to read and write commands. Sami Iren (Seagate) has suggested a mechanism by which the limited number of available bytes in a CDB can be extended to support a scatter/gather list as well as other CDB extensions.

This proposal specifies how to modify OSD-2 in order to achieve both results.

Revision History

r0 Initial revision

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

Proposed Changes in OSD-2

Change 1 – Add & Secure a range of command attributes pages

3.1.f command attributes page: An attributes page (see 4.8.5) that is used to extent the information associated with a CDB (see 3.1.9). See 4.8.5.2.

4.8.4 Command function ordering for commands that get and/or set attributes

...

Commands other than GET ATTRIBUTES, GET MEMBER ATTRIBUTES, SET ATTRIBUTES, SET MEMBER ATTRIBUTES, REMOVE, REMOVE MEMBER OBJECTS, REMOVE PARTITION, and REMOVE COLLECTION that include getting or setting attributes shall be processed in the following order:

- 1) Process any set attributes commands functions specified in the CDB that affect command attributes pages (see 4.8.5);
- 2) Process those command functions not related to attributes (e.g., writing data to a user object);
- 3) Process any set attributes command functions resulting from the processing of the command (e.g., changes due to a WRITE command);
- 4) Process any set attributes command functions specified in the CDB; and
- 5) Process any get attributes command functions specified in the CDB.

A GET ATTRIBUTES command shall be processed in the following order:

- 1) Process any set attributes commands functions specified in the CDB that affect command attributes pages;
- 2) Process any set attributes command functions resulting from the processing of the command (e.g., updating the attributes related timestamps);

- 3) Process any get attributes command functions specified in the CDB; and
- 4) Process any set attributes command functions specified in the CDB.

A GET MEMBER ATTRIBUTES command shall be processed in the following order:

- 1) Process any set attributes commands functions specified in the CDB that affect command attributes pages;
- 2) Process any set attributes command functions resulting from the processing of the command (e.g., updating the attributes related timestamps);
- 3) Process any get attributes command functions specified in the CDB for the user object members of the collection;
- 4) Process any set attributes command functions specified in the CDB for the user object members of the collection;
- 5) Process any get attributes command functions specified in the CDB for the collection; and
- 6) Process any set attributes command functions specified in the CDB for the collection.

A SET ATTRIBUTES command shall be processed in the following order:

- 1) Process any set attributes commands functions specified in the CDB that affect command attributes pages;
- 2) Process any set attributes command functions resulting from the processing of the command (e.g., updating the attributes related timestamps);
- 3) Process any set attributes command functions specified in the CDB; and
- 4) Process any get attributes command functions specified in the CDB.

A SET MEMBER ATTRIBUTES command shall be processed in the following order:

- 1) Process any set attributes commands functions specified in the CDB that affect command attributes pages;
- 2) Process any set attributes command functions resulting from the processing of the command (e.g., updating the attributes related timestamps);
- 3) Process any set attributes command functions specified in the CDB for the user object members of the collection;
- 4) Process any get attributes command functions specified in the CDB for the user object members of the collection;
- 5) Process any set attributes command functions specified in the CDB for the collection; and
- 6) Process any get attributes command functions specified in the CDB for the collection.

A REMOVE command, a REMOVE MEMBER OBJECTS command, a REMOVE PARTITION command, or a REMOVE COLLECTION command that includes getting or setting attributes shall be processed in the following order:

- 1) Process any set attributes commands functions specified in the CDB that affect command attributes pages;
- 2) Process any set attributes command functions specified in the CDB;
- 3) Process any get attributes command functions specified in the CDB;
- 4) Process those command functions not related to attributes; and
- 5) Process any set attributes command functions resulting from the processing of the command (e.g., updating timestamps).

4.8.5 Attributes pages

4.8.5.1 Overview

Each attributes page contains attributes with similar sources or uses. Identifying numbers are assigned to attributes pages with ranges of page numbers (see table 7) indicating the type of OSD object with which an attributes page is associated.

Table 7 — Attributes page numbers

Page Number	OSD object type with which the attributes page is associated
0h to 2FFF FFFFh	User
3000 0000h to 5FFF FFFFh	Partition
6000 0000h to 8FFF FFFFh	Collection
9000 0000h to BFFF FFFFh	Root
C000 0000h to EFFF FFFFh	Reserved
F000 0000h to FFFF FFFEh	Any OSD object type (i.e., root, partition, collection, or user) ^a
FFFF FFFFh	Any OSD object type ^b
^a Attributes page numbers F000 1000h to F000 FFFFh are used for command attributes pages (see 4.8.5.2).	
^b Attributes page number FFFF FFFFh is used to request the retrieval of all attributes pages for a given OSD object type.	

... {{No other changes in 4.8.5, now 4.8.5.1.}}

4.8.5.2 Command attributes pages

Attributes in the attributes pages numbered F000 1000h to F000 FFFFh are used as extensions to the fields in the CDB (see clause 5).

Attribute values in the command attributes pages are defined only for the lifetime (see SAM-4) of the command that sets them.

Attribute values in a command attributes page are only settable. A command that attempts to retrieve attributes in a command attributes page shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set as follows:

- a) If the invalid attribute length is in a CDB field, the additional sense code shall be set to INVALID FIELD IN CDB; or
- b) If the invalid attribute length is in the Data-Out Buffer, the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

...

4.12.4.4 The CMDRSP security method

The CMDRSP security method validates the integrity of the CDB, status, and sense data for each command. The application client computes the CDB REQUEST INTEGRITY CHECK VALUE field (see 5.2.8) contents using:

- a) The algorithm indicated by the attribute in the Root Policy/Security attributes page (see 7.1.2.21) whose attribute number is specified in the capability INTEGRITY CHECK VALUE ALGORITHM field (see 4.12.3);
- ~~b) All the bytes in the CDB with the bytes in the REQUEST INTEGRITY CHECK VALUE field set to zero; and~~
- b) All the following bytes:
 - 1) All the bytes in the CDB with the bytes in the REQUEST INTEGRITY CHECK VALUE field set to zero; and
 - 2) If the CAP bit is set to one and the GET/SET CDBFMT field is set to 11b in the CDB (see 5.2), then all the bytes in the set attributes list identified by the SET ATTRIBUTES LIST LENGTH field and the SET ATTRIBUTES LIST OFFSET field in the CDB (see 5.2.4.4);and
- c) The credential capability key (see 4.12.5.2).

... {{No other changes in 4.12.4.4.}}

...

...

4.12.6.1 Credential validation

The processes described in this subclause do not apply if the CDB SECURITY METHOD field specifies the NOSEC security method (i.e., if the CDB SECURITY METHOD field contains zero).

If the CDB SECURITY METHOD field specifies the CMDRSP security method or the ALLDATA security method, the device server shall validate the CDB REQUEST NONCE field as described in 4.12.7.2.

The device server shall validate the credential associated with a CDB by:

- 1) Reconstructing the credential containing the capability as described in 4.12.6.2;
- 2) Computing the credential integrity check value for the reconstructed credential using the algorithm, inputs, and secret key specified in 4.12.6.3;
- 3) Computing the request integrity check value using:
 - A) The algorithm indicated by the attribute in the Root Policy/Security attributes page (see 7.1.2.21) whose attribute number is specified in the capability INTEGRITY CHECK VALUE ALGORITHM field (see 4.12.3);
 - B) Based on the contents of the CDB SECURITY METHOD field, one of the following arrays of bytes:
 - a) For the CAPKEY security method, the security token (see 4.12.4.3); ~~or~~
 - ~~b) For the CMDRSP security method and the ALLDATA security method, all the bytes in the CDB with the bytes in the REQUEST INTEGRITY CHECK VALUE field set to zero;~~
 - b) For the CMDRSP security method, the following bytes:
 - 1) All the bytes in the CDB with the bytes in the REQUEST INTEGRITY CHECK VALUE field set to zero; and
 - 2) If the CAP bit is set to one and the GET/SET CDBFMT field is set to 11b in the CDB (see 5.2), then all the bytes in the set attributes list identified by the SET ATTRIBUTES LIST LENGTH field and the SET ATTRIBUTES LIST OFFSET field in the CDB (see 5.2.4.4);
 - c) For the ALLDATA security method, all the bytes in the CDB with the bytes in the REQUEST INTEGRITY CHECK VALUE field set to zero;
 - and
 - C) The credential integrity check value computed in step 2) as the secret key;
 - and
 - ... {{No other changes in 4.12.6.1.}}

{{Notes for security reviewers:

- a) Attempts to slip command attributes in without setting the CAP bit are caught and rejected by the test in 5.2.i;
- b) Attempts to set a command attribute using the set attributes options based on CDB fields are covered by the usual CDB integrity checking;
- c) No changes are needed for the NOSEC and CAPKEY security methods because CDB integrity is not assured by OSD in those methods; and
- d) No changes are needed in the ALLDATA security method because all of the parameter data is integrity checked using a different set of security tools.

}}

5.2 Fields commonly used in OSD commands

5.2.1 Overview

OSD commands employ the basic CDB structure shown in 5.1. Within the basic CDB structure, the OSD service action specific fields are organized so that the same field is in the same location in all OSD CDBs (see table 49). OSD service action specific fields that are unique to a small number of CDBs are not shown in this subclause.

Table 49 — OSD service action specific fields

Bit Byte	7	6	5	4	3	2	1	0
10	Reserved			DPO ^a	FUA ^a	ISOLATION (see 5.2.5)		
11	Reserved		GET/SET CDBFMT ^e		Command-specific options			
11	CAP ^b	Reserved	GET/SET CDBFMT ^c		Command specific options			
12	TIMESTAMPS CONTROL (see 5.2.10)							
13	Reserved							
15								
16	(MSB)	PARTITION_ID (see 5.2.7)						(LSB)
23								
24	(MSB)	USER_OBJECT_ID (see 5.2.11)						(LSB)
31								
32	(MSB)	LENGTH (see 5.2.6) or ALLOCATION LENGTH (see 5.2.2)						(LSB)
39								
40	(MSB)	STARTING BYTE ADDRESS (see 5.2.9)						(LSB)
47								
48	Reserved							
51								
52	Get and set attributes parameters ^b							
79								
80	Capability (see 4.11.2.2)							
183								
184	Security parameters (see 5.2.8)							
223								
^a See 5.2.3. ^b See 5.2.i. ^c See 5.2.4.								

...

5.2.i Command attributes present bit

A CAP (command attributes present) bit set to zero indicates that the get and set attributes parameters (see 5.2.3) do not contain any command attributes pages (see 4.8.5.2). A CAP bit set to one indicates that the get and set attributes parameters include a request to set at least attribute value in a command attribute page.

If the device server encounters an request to set an attribute value in a command attributes page and the CAP bit is set to zero, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set as follows:

- a) If the invalid attribute length is in a CDB field, the additional sense code shall be set to INVALID FIELD IN CDB; or
- b) If the invalid attribute length is in the Data-Out Buffer, the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

NOTE x1 - Attempts to retrieve attributes from a command attributes page are handled as described in 4.8.5.2.

...

6 Commands for OSD type devices

{{For every command except those listed in change 2 (i.e., except for CREATE AND WRITE, READ, and WRITE), modify the command definition table as shown in table x1 and add the text below table x1 following the command definition table.}}

Table x1 — J-RANDOM command

Bit Byte	7	6	5	4	3	2	1	0
10	... {{no changes in byte 10}}							
11	Reserved		GET/SET CDBFMT		Command-specific options			
11	CAP	Reserved	GET/SET CDBFMT		Command specific options			
...	... {{no other command definition table changes}}							

...

The contents of the CAP bit are defined in 5.2.i. If the CAP bit is set to one, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

{{The intent is that the above text will be replaced with text like that shown in change 2 when, as, and if command attributes pages are defined that apply to a given command.}}

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.4.

Change 2 – Use a command attributes page to define scatter/gather extensions to read and write**4.11.2.2.2 USER capability object descriptor**

...

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 any of the following is true:

- a) ~~The range of bytes specified by the CDB LENGTH field and STARTING BYTE ADDRESS field in a CREATE AND WRITE command (see 6.5), READ command (see 6.23), or WRITE command (see 6.32) is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field;~~
- a) The command is a CREATE AND WRITE command (see 6.5), READ command (see 6.23), or WRITE command (see 6.32) with the CAP bit (see 5.2.i) set to zero, and the range of bytes specified by the CDB LENGTH field and STARTING BYTE ADDRESS field is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field;
- b) If the command is a CREATE AND WRITE command, READ command, or WRITE command with the CAP bit set to one and the scatter/gather list attribute in the Data Transfer command attributes page (see 7.1.2.d) is set, the device server shall use one of the following tests to determine whether the command is to be terminated:
 - A) The ALLOWED RANGE LENGTH field contains a value other than zero or ALLOWED RANGE STARTING BYTE OFFSET field contains a value other than FFFF FFFF FFFF FFFFh; or
 - B) Any scatter/gather list entry (see 7.1.2.d) indicates user object bytes that are not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field;
- c) The range of bytes specified by the CDB LENGTH field in an APPEND command (see 6.2) and the value in the user object logical length attribute in the User Object Information attributes page (see 7.1.2.11) is not inside the range of bytes specified by the ALLOWED RANGE LENGTH field and ALLOWED RANGE STARTING BYTE OFFSET field;

... {{No other changes in 4.11.2.2.2.}}

6 Commands for OSD type devices

{{For CREATE AND WRITE, READ, and WRITE commands, modify the command definition table as shown in table x2 and add the text below table x2 following the command definition table.}}

Table x2 — ... READ/WRITE command

Bit Byte	7	6	5	4	3	2	1	0
10	... {{no changes in byte 10}}							
11	Reserved		GET/SET CDBFMT		Command specific options			
11	CAP	Reserved	GET/SET CDBFMT		Command specific options			
...	... {{no other command definition table changes}}							

...

The contents of the CAP bit are defined in 5.2.i. If the CAP bit is set to one, the device server shall process the set attributes parameters (see 5.2.4) as if they contain settings for attribute values in the Data Transfer command attributes page (see 7.1.2.d).

{{The intent is that all the applicable command attributes pages be listed (as an a,b,c list if there are more than one) as part of the above paragraph.}}

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.4.

...

7.1.2.1 Attributes pages overview

...

The attributes pages defined by this standard are shown in table 121.

Table 121 — Attributes pages defined by this standard

Page Number	Page Name	Page Format Defined	Support Requirements	Reference
{{no changes before the Root Error Recovery attributes page}}				
R+6h R+7h to R+7Fh	Root Error Recovery Reserved	Yes	Mandatory	7.1.2.25
F000 0000h to FFFF FFFDh F000 2000h F000 2001h F000 2002h to FFFF FFFDh FFFF FFFEh	Reserved Data Transfer command ^a Reserved Current Command	No Yes	Optional Mandatory	7.1.2.d 7.1.2.29
^a This attributes page is a command attributes page (see 4.8.5.2).				

... {{No other changes in 7.1.2.1.}}

...

{{Insert 7.1.2.d immediately before 7.1.2.29.}}

7.1.2.d Data Transfer command attributes page

{{All of 7.1.2.d is new. Color-based change markups suspended.}}

The Current Command attributes page (FFFF 2001h) is a command attributes page (see 4.8.5.2) that shall contain the attributes listed in table x3.

Table x3 — Data Transfer command attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	n	Scatter/gather list	Yes	No
2h to FFFF FFFEh		Reserved	No	

The page identification attribute (number 0h) shall have the format described in 7.1.2.2 with the VENDOR IDENTIFICATION field containing the ASCII characters "INCITS" and the ATTRIBUTES PAGE IDENTIFICATION field containing the ASCII characters "T10 Data Transfer command".

The scatter/gather list attribute (number 1h) indicates the relationship between contents of the command data buffer segment (see 4.14.3 for the Data-In Buffer or 4.14.4 for the Data-Out Buffer) and the bytes in the user object as shown in table x4.

Table x4 — Scatter/gather list attribute format

Bit Byte	7	6	5	4	3	2	1	0
	Scatter/gather list entries							
0	Scatter/gather list entry [first] (see table x5)							
15								
	⋮							
n-15	Scatter/gather list entry [last] (see table x5)							
n								

Each scatter/gather list entry indicates the starting byte offset in the user object and number of bytes to be transferred to or from the command data buffer segment.

The first byte in the command data buffer segment is transferred to or from the user object byte offset indicated by the first scatter/gather list entry. Additional bytes are transferred to or from the byte by byte until the number of bytes indicated by the first scatter/gather list entry have been transferred.

The next byte in the command data buffer segment (i.e., the first byte in the command data buffer segment that was not transferred by the first scatter/gather list entry) is transferred to the user object byte offset indicated by the

second scatter/gather list entry. Additional bytes are transferred to or from the byte by byte until the number of bytes indicated by the second scatter/gather list entry have been transferred.

This process is repeated until all the bytes indicated by the CDB LENGTH field have been transferred or all the scatter/gather list entries have been processed, which ever occurs first.

If the scatter/gather list attribute is set and the CDB STARTING BYTE ADDRESS field is not set to zero, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

Each scatter/gather list entry has the format shown in table x5.

Table x5 — Scatter/gather list entry format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB) _____							
7	USER OBJECT BYTE OFFSET (LSB)							
8	(MSB) _____							
15	BYTES TO TRANSFER (LSB)							

The USER OBJECT BYTE OFFSET field indicates the starting byte offset in the user object for this scatter/gather list entry.

The BYTES TO TRANSFER field indicates the number of bytes to transfer for this scatter/gather list entry.

The byte ranges specified by individual scatter/gather list entries are allowed to overlap (e.g., the second scatter/gather list entry may transfer some or all of the same bytes that were transferred by the first scatter/gather list entry).

If a attributes list (see 5.2.4.4) contains an entry specifying the number of an attribute that table x3 states is not application client settable, 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 PARAMETER LIST. If the CDB SET ATTRIBUTE NUMBER field (see 5.2.4.3) specifies the number of an attribute that table x3 states is not application client settable, 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.

{{Annex B must be updated to include the Data Transfer command attributes page and all the attributes defined therein.}}