ENDL
T E X A S

Date: 7 July 2007
To: T10 Technical Committee
From: Ralph O. Weber
Subject: OSD-2 Attributes Enhacements

Introduction

The SNIA OSD TWG has requested the following changes and enhancements to attributes definitions:
   a) Clarification of defined and undefined attributes;
   b) The ability to set one attribute of length 1 to 18 bytes without consuming bytes in the Data-Out Buffer;
   c) The definition of an attribute that indicates a multi-object command is processing a collection; and
   d) The definition of four attributes that allow the reservation of space for individual user objects or partitions.

Revision History

   r0   Initial revision
   r1   Revised as pre requests from the SNIA OSD TWG
   r2   Added the APPEND command in two places where it serves as an additional flavor of the WRITE command as recommended by Sami Iren

Differences between **r0** and r2 are identified by change bars.

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

Change 1 – Clarify Defined and Undefined Attributes

Problem Statement

Some readers of OSD-2 r01 have concluded that three kinds of attributes are specified: defined attributes, undefined attributes, and attributes with zero length. The intention is that only two kinds of attributes are mentioned in the standard: defined and undefined. Attributes with zero length are the same as undefined attributes.

Proposed OSD-2 r01 changes

{{Add the following new glossary entries in proper alphabetical order.}}

3.1.d defined attribute: An attribute (see 3.1.3) for which the OSD logical unit (see 3.1.27) is storing a value of length greater than zero (i.e., an attribute that is known to the OSD logical unit).

3.1.u undefined attribute: An attribute (see 3.1.3) for which the only attribute length known to the OSD logical unit (see 3.1.27) is zero (i.e., an attribute whose attribute page and attribute number are valid but that is unknown to the OSD logical unit).

…

4.7 OSD object attributes

4.7.1 Overview

… {{4.7.1 is unchanged except for last paragraph and additions that follow it.}}

The structures of attributes pages are ~~defined~~ described by standards (e.g., this standard, other American National Standards, ISO standards), by OSD applications specifications (e.g., SAN file systems, data base systems, fixed data repositories), by publicly available manufacturer specifications, and by other documentation. A range of vendor specific attributes pages is ~~defined~~ provided for which the usage is not restricted by this standard.

Attributes for which the OSD logical unit is able to return a value (i.e., attributes with a non-zero length) are called defined attributes. Attributes with no returnable value (i.e., attributes with a zero length) are called undefined attributes. All the attributes described by this standard are defined attributes.

4.7.2 Attribute length when retrieving defined and undefined attributes

When an attribute length is returned as part of the retrieved attribute information for a defined attribute (see 3.1.d), the attribute length is the non-zero length of the attribute value.

Attempts to retrieve an undefined attribute (see 3.1.u) shall result in an attribute length of zero being returned (see 7.1.3.3). The number of bytes in the attribute value field is zero and six pad bytes are included to eight-byte align the descriptor.

{{Note: OSD and OSD-2 r01 return a length of FFFFh, but this is no longer necessary because a length of zero specifies an undefined attribute. Therefore, zero should indicate an undefined attribute too.}}

4.7.3 Attribute length when setting defined and undefined attributes

The combination the length to which an attribute is being set using CDB set attributes parameters (see 5.2.2) and whether the OSD logical unit knows the attribute as a defined attribute (see 3.1.d) or an undefined attribute (see 3.1.u) affect an attribute setting operation as shown in table x1.

Table x1 — Setting defined attributes and undefined attributes

Attribute length to be set	Defined attribute	Undefined attribute
zero	Change the attribute from being a defined attribute to being an undefined attribute.	Leave the attribute as an undefined attribute. This shall not be considered an error.
non-zero	Replace the current value of the defined attribute with the specified new value. This shall not be considered an error.	Change the attribute from being an undefined attribute to being a defined attribute with the specified value.

If this standard specifies the length of an attribute, any command that attempts to set that attribute's length to a value other than what this standard specifies (e.g., a command that attempts to undefined the user object length attribute described in 7.1.2.11) shall be terminated with a CHECK CONDITION, 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.

4.7.4 ~~4.7.2~~ Command function ordering for commands that get and/or set attributes

…

4.7.5 ~~4.7.3~~ Attributes pages

…

Except for the attributes page numbers that apply to any OSD object type (i.e., F000 0000h through FFFF FFFFh), the ranges of attributes page numbers shown in table 6 are subdivided as shown in table 7.

Table 7 — Attributes page number sets

Page Number Within Range	Description
0h to 7Fh	Defined by this standard
80h to 7FFFh	Reserved
8000h to EFFFh	Defined by other standards (see Annex A)
F000h to FFFFh	Defined by publicly available manufacturer specifications
1 0000h to 1FFF FFFFh	Assigned by the OSD logical unit [a]
2000 0000h to 2FFF FFFFh	Vendor specific

> [a] The attributes in these pages are ~~not defined~~ undefined (see 3.1.u) until they are set (see 4.7.5) ~~using CDB set attributes parameters (see 5.2.2)~~. The attribute number 0h should be set as specified in 7.1.2.2 to maintain correct attribute directory information. ~~The attributes in these pages may be set repeatedly and the OSD logical unit shall maintain the most recently set values for retrieval using the CDB get attributes parameters.~~ The OSD logical unit shall not modify attribute values in these pages except in response to information provided in the set attributes parameters in a CDB.

…

4.7.6 ~~4.7.4~~ Attributes

Each attribute within an attributes page (see 4.7.5) has a unique number between 0h and FFFF FFFEh. The description of each attribute defines the format and usage of that attribute. For examples of attribute definitions see 7.1.2.

The attribute with the attribute number 0h contains the name of the page in the format described in 7.1.2.2. The attribute number FFFF FFFFh may be used to request the retrieval of all the defined attributes (see 3.1.d) in a page ~~having a non-zero attribute length (see 7.1.3)~~.

~~If an attributes list specifies the setting of~~ If a command attempts to set attribute number FFFF FFFFh, ~~the command~~ it shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set ~~to INVALID FIELD IN PARAMETER LIST.~~ as follows:

    a)   If the invalid attribute number is in a CDB field, the additional sense code shall be set to INVALID FIELD IN CDB; or

    b)   If the invalid attribute number is in the Data-Out Buffer, the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

4.7.7 4.7.5 Attributes directories

…

7.1.1 Attributes parameter formats

…

A get attributes request for an attribute or attributes page having no previously established value shall not be considered an error. If an attribute value that has not been previously established is requested by specific attribute number, a list entry format value (see 7.1.3.3) (see 7.1.3) having FFFF FFFFh in with the ATTRIBUTE LENGTH field set to zero (see 4.7.2) shall be returned. If an attributes page that has no established definition is requested, a null attributes page (see 7.1.2.25) shall be returned.

…

7.1.3 OSD attributes lists

7.1.3.1 Attributes lists overview

…

For an attributes list sent from the device server to the application client, a list length of zero indicates that all of the requested attributes are undefined attributes (see 3.1.u). have an attribute length of zero.

…

7.1.3.2 List entry format for retrieving attributes for the specified OSD object

…

The ATTRIBUTE NUMBER field specifies:

   a) The attribute number within the attributes page specified by the ATTRIBUTES PAGE field of the one attribute value to be returned; or
   b) The value FFFF FFFFh to request the return of each defined attribute (see 3.1.d) having a non-zero attribute length in the attributes page specified by the ATTRIBUTES PAGE field.

Requirements on the attribute length field for retrieved attributes are described in 4.7.2.

If the attribute specified by the ATTRIBUTES PAGE field and ATTRIBUTE NUMBER field has no defined value, an attribute value having a length of FFFFh shall be returned.

…

If FFFF FFFFh is used as an attributes page number or attribute number value, only those defined attributes (see 3.1.d) with non-zero lengths shall be returned.

7.1.3.3 List entry format for retrieved attributes and for setting attributes for the specified OSD object

…

The ATTRIBUTE LENGTH field specifies the length of the attribute value in bytes. The contents of the ATTRIBUTE LENGTH field and ATTRIBUTE VALUE field for retrieved attributes are described in 4.7.2. The effects of the ATTRIBUTE LENGTH field on the setting of attributes are described in 4.7.3.

The ATTRIBUTE VALUE field specifies the attribute value.

If the attribute specified by the ATTRIBUTES PAGE field and ATTRIBUTE NUMBER field in a set command function has a defined value, the value shall be replaced by the value specified by the ATTRIBUTE LENGTH field and ATTRIBUTE VALUE field. Otherwise, a new attribute shall be created with the attribute number specified by the ATTRIBUTE NUMBER field in the attributes page specified by the ATTRIBUTES PAGE field.

…

7.1.3.4 List entry format for attributes retrieved by CREATE command that creates multiple user objects

…

The ATTRIBUTE LENGTH field specifies the length of the attribute value in bytes. The contents of the ATTRIBUTE LENGTH field and ATTRIBUTE VALUE field for retrieved attributes are described in 4.7.2.

… {{No other changes in 7.1.3.}}

Change 2 – Setting One Short Attribute

Problem Statement

Currently, setting one attribute requires placing bytes in the Data-Out Buffer. Allocating, pinning, and mapping of buffers is expensive, and this expense is hardly worthwhile for setting a 4 or 8 byte attribute value. OSD-2 needs a simple way to set one attribute and it is proposed to place one attribute value setting value in the CDB.

Proposed OSD-2 r01 changes

5.2.2 Get and set attributes parameters

5.2.2.1 Get and set attributes CDB format selection

The GET/SET CDBFMT (get and set attributes CDB format) field (see table 45) specifies the format of the get and set attributes parameters in the CDB.

Table 45 — Get and set attributes CDB format code values

Value	Description	Reference
00b ~~01b~~	Reserved	
01b	Set one attribute using CDB fields	5.2.2.2
10b	Get an attributes page and set an attribute value	5.2.2.~2~3
11b	Get and set attributes using lists	5.2.2.~3~4

{{No other changes in 5.2.2.1.}}

5.2.2.2 Set one attribute value using CDB fields

{{All of 5.2.2.2 is new. Text change markups suspended.}}

The set one attribute using CDB parameters format (see table x2) allows the setting of a single attribute using CDB fields and does not allow the retrieval of any attributes.

Table x2 — Set one attribute using CDB parameters format

Bit Byte	7	6	5	4	3	2	1	0
⋮ 51				Other CDB fields				
52 55	(MSB)			ATTRIBUTES PAGE				(LSB)
56 59	(MSB)			ATTRIBUTE NUMBER				(LSB)
60 61	(MSB)			ATTRIBUTE LENGTH				(LSB)
62 79	(MSB)			ATTRIBUTE VALUE				(LSB)
80				⋮ Other CDB fields				

The ATTRIBUTES PAGE field specifies the page number of the attribute value to be set.

The ATTRIBUTE NUMBER field specifies the attribute number within the attributes page specified by the ATTRIBUTES PAGE field of the attribute value to be set.

If the ATTRIBUTES PAGE field or ATTRIBUTE NUMBER field contains FFFF FFFFh, the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

The ATTRIBUTE LENGTH field specifies the length of the attribute value in bytes. If the attribute length is greater than 18, the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the ATTRIBUTE LENGTH field does not contain zero, the ATTRIBUTE VALUE field specifies the attribute value. If the ATTRIBUTE LENGTH field contains zero, the contents of the ATTRIBUTE VALUE field are ignored. The attribute length field shall affect the setting of the attribute as described in 4.7.3.

If setting an attribute value 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.

{{Other than subclause numbers, there are no changes in 5.2.2.2 and 5.2.2.3.}}

Change 3 – Operation In Progress attribute

Problem Statement

It is desirable to have an attribute that indicates an active multi-object operation is in progress.

Proposed OSD-2 r01 changes

4.6.6.2 Commands that use collections to affect multiple user objects

...{{07-270 also modifies the following text as shown in orange.}}

If a multi-object command is terminated as part of processing any task management function or as the result of a condition (e.g., logical unit reset) established in response to an event (see SAM-4), then the device server shall either:

   a) Set the multi-object operation in progress attribute value to zero in the Collection Information attributes page (see 7.1.2.10); and
   b) Either:
      A) Establish a consistent, stable state for each user object being processed; or
      B) Set the policy access tag attribute in the User Object Policy/Security attributes page described in 4.9.3 for any user object for which it is not possible to establish consistent state.

...{{insert the following near the end of 4.6.6.2}}

The multi-object operation in progress attribute in the Collection Information attributes page (see 7.2.1.10) shall be set as follows:

   a) To one before an operation is performed as described in this subclause on the first user object in a collection; and
   b) To zero before the processing of a multi-object command is completed or terminated as described in this subclause.

7.1.2.10 Collection Information attributes page

The Collection Information attributes page (C+1h) shall contain the attributes listed in table 116.

Table 116 — Collection Information attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	8	Partition_ID	No	Yes
2h	8	Collection_Object_ID	No	Yes
3h to 8h		Reserved	No	
9h	variable	Username	Yes	No
Ah	1	Collection type	No	Yes
Bh	4	Number of members	No	Yes
Ch	1	Multi-object operation in progress	No	Yes
~~C~~Dh to 80h		Reserved	No	
81h	8	Used capacity	No	Yes
82h to FFFF FFFEh		Reserved	No	

…

The multi-object operation in progress attribute (Ch) shall contain the following:

    a)   Zero if no multi-object operations (see 4.6.6.2) are in progress; or
    b)   One if a multi-object operation is in progress.

If a multi-object command is terminated as part of processing any task management function or as the result of a condition (e.g., logical unit reset) established in response to an event (see SAM-4), then the device server shall set the multi-object operation in progress attribute value to zero.

… {{no other changes in 7.1.2.10.}}

Change 4 – Space Reservations Attributes

Problem Statement

Application clients need a way to guarantee that space will be available when previously written user data (i.e., excluding meta data) is flushed. To address this need, two new attributes are proposed for the User Object Information attributes page.

Proposed OSD-2 r01 changes

7.1.2.11 User Object Information attributes page

The User Object Information attributes page (1h) shall contain the attributes listed in table 118.

Table 118 — User Object Information attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	8	Partition_ID	No	Yes
2h	8	User_Object_ID	No	Yes
3h to 8h		Reserved	No	
9h	variable	Username	Yes	No
Ah to 80h		Reserved	No	
81h	8	Used capacity	No	Yes
82h	8	User object logical length	Yes	Yes
83h to D0h		Reserved	No	
D1h	0 or 8	Actual data space	No	Yes
D2h	0 or 8	Reserved data space	Yes	No
83h D3h to FFFF FFFEh		Reserved	No	

…

The used capacity attribute (number 81h) shall contain the sum of: ~~the number of bytes used by the user object including attributes bytes.~~

  a)   The number of bytes used by the user object including attributes bytes; and
  b)   The number of unused reserved bytes computed as:
    A)   Value in the reserved data space attribute in this User Object Information attributes page minus the value in the actual data space attribute in this User Object Information attributes page; or
    B)   Zero if the value in the actual data space attribute in this User Object Information attributes page is larger than the value in the reserved data space attribute in this User Object Information attributes page.

…

If the OSD logical unit does not support the reserved data space attribute, the actual data space attribute (D1h) shall be undefined (see 3.1.u). If the reserved data space attribute is supported, the actual data space attribute shall be defined (see 3.1.d) and shall contain the number of bytes used by the user object to store data transferred in the command data or parameter data segment of the Data-Out Buffer (see 4.12.4) by APPEND commands (see 6.2), CREATE AND WRITE commands (see 6.4), and/or WRITE commands (see 6.28) to the user object.

If the reserved data space attribute (D2h) is defined (see 3.1.d), it contains the minimum value that the actual data space attribute in this User Object Information attributes page is allowed to reach before an APPEND command, a CREATE AND WRITE command, or a WRITE command may be terminated with the ADDITIONAL SENSE CODE field set to 55h (i.e., SYSTEM RESOURCE FAILURE).

… {{No other changes in 7.1.2.11.}}

7.1.2.9 Partition Information attributes page

The Partition Information attributes page (P+1h) shall contain the attributes listed in table 115.

Table 115 — Partition Information attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	8	Partition_ID	No	Yes
2h to 8h		Reserved	No	
9h	variable	Username	Yes	No
Ah to 80h		Reserved	No	
81h	8	Used capacity	No	Yes
82h to C0h		Reserved	No	
C1h	8	Number of collections and user objects	No	Yes
C2 to D0h		Reserved	No	
D1h	0 or 8	Actual data space	No	Yes
D2h	0 or 8	Reserved data space	Yes	No
~~C2h~~ D3h to FFFF FFFEh		Reserved	No	

…

For all partitions except partition zero, the used capacity attribute (number 81h) shall contain the number of allocated bytes for ~~used by~~ the partition as described in this subclause. ~~, all collections, and all user objects within the partition including attributes bytes.~~ For partition zero, the used capacity attribute shall contain the number of allocated bytes for ~~used by~~ partition zero and all other partitions as described in this subclause. ~~, all collections, and all user objects within all partitions including attributes bytes.~~ The number of allocated bytes shall be computed as follows:

   a) The number of bytes used by:
       A) The partition or partitions;
       B) All collections within the partition or partitions; and
       C) All user objects within the partition or partitions including attributes bytes;
       and
   b) The number of unused reserved bytes computed as:
       A) Value in the reserved data space attribute in this Partition Information attributes page minus the value in the actual data space attribute in this Partition Information attributes page; or
       B) Zero if the value in the actual data space attribute in this Partition Information attributes page is larger than the value in the reserved data space attribute in this Partition Information attributes page.

…

If the OSD logical unit does not support the reserved data space attribute, the actual data space attribute (D1h) shall be undefined (see 3.1.u). If the reserved data space attribute is supported, the actual data space attribute shall be defined (see 3.1.d) and shall contain the number of bytes used by all user objects in the partition to store data transferred in the command data or parameter data segment of the Data-Out Buffer (see 4.12.4) by APPEND commands (see 6.2), CREATE AND WRITE commands (see 6.4), and/or WRITE commands (see 6.28) to a user object.

If the reserved data space attribute (D2h) is defined (see 3.1.d), it contains the minimum value that the actual data space attribute in this Partition Information attributes page is allowed to reach before an APPEND command, a CREATE AND WRITE command, or a WRITE command may be terminated with the ADDITIONAL SENSE CODE field set to 55h (i.e., SYSTEM RESOURCE FAILURE).

… {{No other changes in 7.1.2.9.}}

{{Annex C must be updated to add all new attributes defined by this proposal.}}