ENDL
TEXAS

Date: 15 September 2007
To: T10 Technical Committee & SNIA OSD TWG
From: Ralph O. Weber
Subject: OSD-2 Exceptions Management enhancements

Introduction

The SNIA OSD TWG has developed several OSD-2 enhancements to effect a greater level of initiator-based error recovery. This proposal adapts those enhancements to the SCSI device environment and aligns them with the existing OSD command set as defined in OSD-2 r02.

Revision History

r0 Initial revision

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

The proposed changes are in order by subclause. Several additions to the model clause provide an overview of the command set and attributes pages changes that follow. Some entire subclauses are new and the additions, ~~deletions~~, and comments notations described above are suspended in these subclauses.

Proposed Changes in OSD-2

3.1 Definitions

3.1.54 vendor specific (VS): Something (e.g., a bit, field, code value, behavior) that is not defined by this standard and may be vendor defined.

3.2 Acronyms

{{Add the following in alphabetical order.}}

/ division
VS Vendor Specific (see 3.1.54)

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

4.6 Stored data objects

…

4.7 Data object accessibility

{{all of 4.7 is new; text markups suspended}}

4.7.1 OSD logical unit managed accessibility

An OSD logical unit may make some or all of the OSBD storage inaccessible as described in 4.11.3.3 and in 6.j.

The root structure check accessibility attribute in the Root Always Accessible attributes page (see 7.1.2.y) indicates whether the root object and all partitions are accessible. The partition structure check accessibility attribute in the Partition Always Accessible attributes page (see 7.1.2.z) indicates whether a partition is accessible. The error reporting that applies when these attributes indicate one or more objects are inaccessible is described in 6.j.

Editors Note 1 - ROW: The inaccessibility case described in 4.11.3.3 is not covered by the SNIA OSD TWG, but it seems clearly necessary.
In my opinion, the attributes described in this subclause are unneeded as a result of the typical SCSI command processing requirements stated in 4.11.3.3 and in 6.j. I believe the attributes (and this subclause) should be removed from the proposal, but this has yet to be agreed to by the SNIA OSD TWG.

4.7.2 Application client managed accessibility

Write access to user object data (see 4.6.5), collection membership (see 4.6.6), the creation of new user object, the creation of new collections, the creation of new partitions, or the attributes (see 4.7) associated with any type of object, may be controlled using:

a) Policy/storage manager capabilities (see 4.11.2); or
b) Object accessibility attributes in the following attributes pages:
   A) The User Object Information attributes page (see 7.1.2.11);
   B) The Collection Information attributes page (see 7.1.2.10);
   C) The Partition Information attributes page (see 7.1.2.9); and
   D) The Root Information attributes page (see 7.1.2.8).

The object accessibility attributes form the following prioritized hierarchy:

1) Root (i.e., the root object, all partitions, all collections, and all user objects);
2) Partition (i.e., all partition attributes, all collections, and all user objects); and
3) A leaf object (i.e., one collection or one user object).

Denial of write access at a higher level in the hierarchy includes denial of access in all lower levels (e.g., write access to a collection depends on write access being allowed for the collection, the partition, and the root object).

Denial of write access to an object with members means denial of the ability to create new members in that object (e.g., denial of write access to the root object means denial of the ability to create new partitions).

If a command attempts a write access that is not allowed by all applicable levels of the object accessibility attributes hierarchy, then:

a) No part of the command shall be processed; and
b) The command shall be terminated with a CHECK CONDITION status, with the sense key set to DATA PROTECT, the additional sense code set to CONDITIONAL WRITE PROTECT, and the INFORMATION field set as shown in table x1.

Table x1 — DATA PROTECT INFORMATION field contents

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
5								
6	ATTRIBUTE	Reserved						
7	OBJECT TYPE (see table 12)							

If the requested write access that is denied is to the data in a user object, the membership list in a collection, the membership list in a partition, or the membership list in the root object, then the ATTRIBUTE bit shall be set to zero. If the requested write access that is denied in response to an attempt to set an attribute, the ATTRIBUTE bit shall be set to one.

The OBJECT TYPE field shall be set to the object type shown in table 12 (see 4.9.2.2) of the highest member of the object accessibility attributes hierarchy that is denying the requested write access.

{{Additional sense code CONDITIONAL WRITE PROTECT is defined in SPC-4 but not identified for OSD-2 usage.}}

4.8 ~~4.7~~ OSD object attributes

…

4.9 Command atomicity and isolation

{{all of 4.9 is new; text markups suspended}}

4.9.1 Overview

Atomicity refers to the number of bytes that the OSBD writes to stable storage (see 4.11) as a group in a manner that ensures that all the bytes or none of them are written (e.g., in a traditional block storage device the number of bytes of atomicity is one block or 512 bytes).

Isolation refers to the degree of interaction between concurrent commands. The SAM-4 Task Attribute provides a minimal degree of isolation, but it is not sufficient for the complex commands defined in this standard.

Atomicity and isolation interact when atomicity byte count affects the degree of isolation (e.g., if the RANGE isolation method described in table x12 (see 7.1.2.8) uses the data atomicity guarantee to identify which commands overlap for isolation purposes).

4.9.2 Atomicity

The atomicity guarantees described in this standard apply to the following (in priority order):

1) Commands being processed for which GOOD status has not yet been returned; and
2) Commands for which GOOD status has been returned, with the caveat that an application client has no way to distinguish between atomicity effects and effects caused by media failures after it receives the GOOD status.

If a media error is detected while a command is being processed, the atomicity guarantees affect how much data may have been transferred before the error was detected.

If data transfer errors (e.g., media errors, NVRAM failures, power loss) cause a data loss after GOOD status has been returned, then the atomicity guarantees provide one of the boundaries for which data may have been lost. Detecting and recovering from such errors is described in 4.11.3.

Atomicity also affects performance, but control over these effects is outside the scope of this standard.

Application clients use the values in the Root Information attributes page (see 7.1.2.8) attributes shown in table x2 to tailor the commands they send with respect to the atomicity properties of the OBSD.

Table x2 — Atomicity attributes

Attribute	Name	Description
Data atomicity guarantee	D_LIMIT	The minimum number of D_ALIGN aligned user data bytes that the device server shall write to stable storage as a group.
Data atomicity alignment	D_ALIGN	The data alignment value that maximizes the D_LIMIT number of user data bytes that the device server shall write to stable storage as a group. If D_ALIGN is set to zero, then it is processed as if it is set to one.
Attributes atomicity guarantee	A_LIMIT	The minimum number of bytes of an application client settable attribute that the device server shall write to stable storage as a group.
Data/attributes atomicity multiplier	DA_MULT	The multiplier applied to the sum of D_LIMIT and A_LIMIT to determine the minimum number of combined user data and application client settable bytes that the device server shall write to stable storage as a group in response to one command. If either D_LIMIT or A_LIMIT is set to zero, then DA_MULT shall be processed as if it is set to zero.

D_LIMIT and D_ALIGN combine as shown in the following formula to indicate the minimum number of user data bytes that the device server guarantees to write to stable storage as a group:

min_bytes = D_LIMIT - the_remainder_from(starting byte offset / D_ALIGN)

A_LIMIT has no effect on how OSD logical unit provided attributes are written to stable storage. All OSD logical unit provided attributes shall be written to stable storage in a manner that maximizes their integrity and consistency.

Whether the bytes in two or more attributes are written to stable storage as a group is outside the scope of this standard.

If one command writes user data and sets attributes, the D_LIMIT, D_ALIGN, and A_LIMIT attributes still apply, and the minimum total number of bytes that the device server guarantees to write to stable storage as a group is computed as follows:

min_tot_bytes = DA_MULT * ((D_LIMIT - modulo(starting byte offset, D_ALIGN) + A_LIMIT)

Bytes are written to stable storage in multiple groups only when their numbers exceed the guaranteed minimums.

Table x3 shows examples of atomicity attributes effects.

Table x3 — Examples of atomicity attributes effects

Atomicity attribute values				
D_LIMIT	**D_ALIGN**	**A_LIMIT**	**DA_MULT**	**Effect** [a]
0	1	0	0	The device server provides no guarantees regarding the number of bytes written to stable storage as a group.
512	512	0	0	At least 512 bytes of user data aligned on a 512-byte boundary are written to stable storage as a group.
512	512	512	0	At least 512 bytes of user data aligned on a 512-byte boundary are written to stable storage as a group. At least 512 bytes of an application client settable attribute are written to stable storage as a group. The user data may be written in a separate group from the attribute.
512	512	512	1	At least 512 bytes of user data aligned on a 512-byte boundary at least 512 bytes of an application client settable attribute are written to stable storage in one group.

[a] The effects shown in this example all assume that more than 512 bytes of user data and more than 512 bytes of an attribute value are requested to be written.

4.9.3 Isolation

This standard defines several isolation methods in table x12 (see 7.1.2.8).

The default isolation method attribute in the Root Information attributes page (see 7.1.2.8) specifies the isolation method used by commands that do not override the default. Commands may override the default isolation method by specifying a non-zero value in the ISOLATION field.

4.10 ~~4.8~~ Quotas

…

4.11 ~~4.9~~ Policy/storage management

4.11.1 ~~4.9.1~~ Overview

The policy/storage manager:

a) Provides access policy controls to application clients via preparation of policy-coordinated capabilities (see 4.11.2 ~~4.9.2~~); and
b) In concert with the OSD logical unit:~~,~~
   A) Identifies damaged storage within the OBSD (see 4.11.3.2);
   B) Repairs damage that the OSD logical unit is unable to repair without assistance (see 4.11.3.2); and
   C) ~~prevents~~ Uses access policy controls to prevent unsafe or temporarily undesirable utilization of OBSD storage (see 4.11.3.2 ~~4.9.3~~).

…

4.11.2 ~~4.9.2~~ Capabilities

…

4.11.2.2 ~~4.9.2.2~~ Capability format

4.11.2.2.1 ~~4.9.2.2.1~~ Introduction

…

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
…	…	…	…
An OBJECT STRUCTURE CHECK command addressed to a partition	PARTITION	DEV_MGMT	PAR
An OBJECT STRUCTURE CHECK command addressed to the root object	ROOT	DEV_MGMT	PAR
…	…	…	…
A READ MAP command	USER	DEV_MGMT	USER
…	…	…	…
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.			

…

4.11.3 OBSD storage damage detection, repair, and undesirable utilization prevention ~~4.9.3 Policy access tags~~

4.11.3.1 Normal usage storage damage detection and repair {{all of 4.11.3.1 is new; text markups suspended}}

The OBSD device server detects damaged storage when:

a) An application client initiated operation detects an uncorrectable error; or
b) A background operation outside the scope of this standard detects an uncorrectable error.

When a device server detects uncorrectable storage damage, it does the following:

a) Sets the FENCE bit to one as described in 4.11.3.2 in the affected objects;
b) Summarizes the damage by updating the attributes in the Error Recovery attributes pages of the affected objects (e.g., the User Object Error Recovery attributes page (see 7.1.2.x) is updated if the OSD object is a user object); and
c) Establishes a unit attention condition (see SAM-4) for the initiator port associated with every I_T nexus except the I_T nexus, if any, on which the storage damage has caused a CHECK CONDITION status to be returned as follows:
    A) If the storage damage affects some, but not all, partitions, then a unit attention condition shall be established for each affected partition with the:
        a) Sense key set to UNIT ATTENTION;
        b) Additional sense code set to ERROR RECOVERY ATTRIBUTES HAVE CHANGED; and
        c) INFORMATION field set to the Partition_ID of an affected partition;

    or
    B) If the storage damage affects the root object or all partitions, then a unit attention condition shall be established affected partition with the:
        a) Sense key set to UNIT ATTENTION;
        b) Additional sense code set to ERROR RECOVERY ATTRIBUTES HAVE CHANGED; and
        c) INFORMATION field set to zero.

{{ERROR RECOVERY ATTRIBUTES HAVE CHANGED is a new additional sense code. An ASC field value of 2Ah is recommended.}}

An application client that receives notification of uncorrectable damaged storage should forward the notification to the policy/storage manager.

A policy/storage manager that receives notification of uncorrectable damaged storage should:

a) Use any information received with the notification and appropriate commands (e.g., the QUERY command (see 6.20), the GET ATTRIBUTES command (see 6.13), the READ MAP command (see 6.j)) to identify appropriate repair actions;
b) Perform the identified repair actions (e.g., rewrite corrupted data using the WRITE command (see 6.30));
c) Update the affected Error Recovery attributes page or pages; and
d) Set the affected FENCE bits to zero (see 4.11.3.2).

Application clients and policy/storage managers also may detect data errors that are invisible to the device server. Utilization of such data may be prevented by changing the VERSION field in the policy access tag attributes in the Policy/Security attributes pages associated with the affected objects (e.g., the User Object Policy/Security attributes page (see 7.1.2.23) if the OSD object is a user object) as described in 4.11.3.2.

During normal operation, the policy/storage manager may use the OBJECT STRUCTURE CHECK command (see 6.j) to validate the OBSD storage structures for a partition or the root object (i.e., the entire OBSD), however, the

object structures being validated by the OBJECT STRUCTURE CHECK command are inaccessible as described in 6.j for the duration of command processing.

The device server shall not require the application client to send an OBJECT STRUCTURE CHECK command except as described in 4.11.3.3.

4.11.3.2 ~~4.9.3~~ Policy access tags

{{The only change in this subclause is the addition of a cross reference in the second paragraph after table 20. The entire OSD-2 r02 text is included to facilitate reviewing the old text in the context of the new error recovery features.}}

The policy access tag (see table 20) allows the coordinated actions of both the OSD logical unit and policy/storage manager to prevent unsafe or temporarily undesirable utilization of OBSD storage that is assigned to the OSD logical unit.

Table 20 — Policy access tag format

Bit / Byte	7	6	5	4	3	2	1	0
0	FENCE	(MSB)						
1								
2				VERSION				
3								(LSB)

During normal operation the value of the FENCE bit is zero.

If the OSD logical unit detects a condition that would make further accesses to one or more OSD objects unsafe, it shall set the FENCE bit to one in the policy access tag attributes in the Policy/Security attributes pages associated with those objects (e.g., the User Object Policy/Security attributes page (see 7.1.2.23) if the OSD object is a user object) and notify the policy/storage manager of a condition needing attention (see 4.11.3.1). The OSD logical unit, policy/storage manager, or both act to correct whatever conditions are making accesses to the OSD objects unsafe. After the conditions making accesses to the OSD objects unsafe are corrected the policy/storage manager sets the FENCE bit to zero.

If a set attributes list (see 5.2.3.4) contains a request to set the FENCE bit to one, 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 contains 4000 0001h (i.e., the policy access tag attribute) and the set attributes data specified by the SET ATTRIBUTES OFFSET field (see 5.2.3.3) specifies that the FENCE bit be set to one, 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.

To block capability-based access to one or more OSD objects, the policy/storage manager changes the VERSION field in the policy access tag attributes in the Policy/Security attributes pages associated with those objects. The conditions under which the policy/storage manager may be called on to do this include:

a) Recovery from errors other than those detected by the OSD logical unit that make accesses to one or more OSD object unsafe; and
b) Receipt of a request to change the policy access tag from the security manager (see 4.10.6.4).

If a set attributes list (see 5.2.3.4) contains a request to set the VERSION field to zero, 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 contains 4000 0001h (i.e., the policy access tag attribute) and the set attributes data specified by the SET ATTRIBUTES OFFSET field (see 5.2.3.3) specifies that the VERSION field be set to zero, 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.

The OSD logical unit shall not modify the contents of a policy access tag VERSION field.

The device server terminates any command received with a capability whose POLICY ACCESS TAG field contains a non-zero value that differs from the policy access tag attribute value in the Policy/Security attributes page associated with the object (see 4.9.2.2).

4.11.3.3 Storage damage detection and repair after a reset {{all of 4.11.3.3 is new; text markups suspended}}

After a hard reset SCSI device condition established in response to an event (see SAM-4), the device server may oblige the application client to send an OBJECT STRUCTURE CHECK command (see 6.j) for:

a) One or more individual partitions; or
b) The root object and all partitions.

If after a hard reset the device server has determined that processing of an OBJECT STRUCTURE CHECK command for the root object and all partitions is necessary to ensure proper OBSD storage integrity, then it shall:

a) Terminate all received commands except INQUIRY, REPORT LUNS, and REQUST SENSE with a CHECK CONDITION status, with the sense key set to NOT READY, the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED, and the INFORMATION field set to zero; and
b) Complete received REQUEST SENSE commands with GOOD status, with the sense key set to NOT READY, the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED, and the INFORMATION field set to zero.

The need to process an OBJECT STRUCTURE CHECK command for the root object and all partitions shall not affect the processing of the INQUIRY command and REPORT LUNS command.

If after a hard reset the device server has determined that processing of an OBJECT STRUCTURE CHECK command for a partition is necessary to ensure proper OBSD storage integrity, then it shall terminate all received commands addressed to that partition with a CHECK CONDITION status, with the sense key set to NOT READY, the additional sense code set to LOGICAL UNIT NOT READY, INITIALIZING COMMAND REQUIRED, and the INFORMATION field set to the Partition_ID of a partition for which the processing of an OBJECT STRUCTURE CHECK command is needed.

The status and sense data described in this subclause is returned for all received commands until a suitable OBJECT STRUCTURE COMMAND command has begun processing.

4.12 ~~4.10~~ Security

…

~~4.13 Interactions between concurrently processed commands~~

~~The interactions between commands that the device server processes concurrently may be modified using fields in the Control mode page (see SPC-3). This standard defines no other restrictions on the interactions between concurrently processed commands.~~

~~Application clients should ensure that the device server is not requested to process two or more commands concurrently if the interactions between those commands might adversely affect the information returned to the application client by future commands.~~

4.17 ~~4.15~~ Reservations

…

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
…	…	…	…	…	…
OBJECT STRUCTURE CHECK	Conflict	Conflict	Allowed	Conflict	Conflict
…	…	…	…	…	…
READ MAP	Conflict	Conflict	Allowed	Conflict	Conflict
…	…	…	…	…	…
Key: **Excl**=Exclusive, **RR**=Registrants Only or All Registrants					

5 Common Formats

5.1 OSD CDB format

…

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 44). OSD service action specific fields that are unique to a small number of CDBs are not shown in this subclause.

Table 44 — OSD service action specific fields

Bit Byte	7	6	5	4	3	2	1	0
~~10~~	~~OPTIONS BYTE (see 5.2.5)~~							
10	Reserved			DPO [a]	FUA [a]	ISOLATION (see 5.2.y)		
11	Reserved		GET/SET CDBFMT [b]		Command specific options			
12	TIMESTAMPS CONTROL (see 5.2.9)							
13 15	Reserved							
16 23	(MSB) PARTITION_ID (see 5.2.6)							(LSB)
24 31	(MSB) USER_OBJECT_ID (see 5.2.10)							(LSB)
32 39	(MSB) LENGTH (see 5.2.4)							(LSB)
40 47	(MSB) STARTING BYTE ADDRESS (see 5.2.8)							(LSB)
48 51	Reserved							
52 79	Get and set attributes parameters [b]							
80 159	Capability (see 4.9.2.2)							
160 199	Security parameters (see 5.2.7)							

[a] See 5.2.x. {{Note: DPO and FUA are the only options byte bits defined by OSD or OSD-2 r02.}}
[b] See 5.2.3.

5.2.2 Allocation length

…

~~5.2.5 Options byte~~

~~The options byte is a set of fields (see table 49) used to modify or control the command.~~

~~Table 49 — Option byte format~~ {{Completely delete table 49.}}

~~Bit~~	~~7~~	~~6~~	~~5~~	~~4~~	~~3~~	~~2~~	~~1~~	~~0~~
	~~Reserved~~			~~DPO~~	~~FUA~~	~~Reserved~~		

~~The DPO (disable page out) bit allows the application client to influence the use of volatile cache (see 4.11). If the DPO bit is set to zero, the use of volatile cache should proceed without influence caused by the DPO bit value. If the DPO bit is set to one, the device server should not place data transferred as a result of this command in the volatile cache.~~

~~The FUA (force unit access) bit controls whether or not the results of a command shall be written to stable storage (see 4.11) before status is returned to the application client. If the FUA bit is set to zero, the device server may return status as soon as the data transferred by this command is in the volatile cache. If the FUA bit is set to one, the device server shall not return status until the data transferred by this command (i.e., either read data or write data) has been written to stable storage.~~

~~The direction of data transfer has no effect on the meaning of the DPO and FUA bits. The DPO and FUA bits affect the processing of both OSD object data and attributes.~~

{{Add the following subclauses to 5.2 in the proper alphabetical order by field name.}}

5.2.x Caching control bits

{{all of 5.2.x is new; text markups suspended}}

The DPO (disable page out) bit allows the application client to influence the use of volatile cache (see 4.11). If the DPO bit is set to zero, the use of volatile cache should proceed without influence caused by the DPO bit value. If the DPO bit is set to one, the device server should not place data transferred as a result of this command in the volatile cache.

The FUA (force unit access) bit controls whether or not the results of a command shall be written to stable storage (see 4.11) before status is returned to the application client. If the FUA bit is set to zero, the device server may return status as soon as the data transferred by this command is in the volatile cache. If the FUA bit is set to one, the device server shall not return status until the data transferred by this command (i.e., either read data or write data) has been written to stable storage.

The direction of data transfer has no effect on the meaning of the DPO and FUA bits. The DPO and FUA bits affect the processing of both OSD object data and attributes.

5.2.y Isolation

{{all of 5.2.y is new; text markups suspended}}

The ISOLATION field (see table x4) specifies the isolation mode to be applied to this command.

Table x4 — ISOLATION field

Code	Description
0h	The isolation method specified by the default isolation method attribute in the Root Information attributes page (see 7.1.2.8) shall be applied to this command.
1h	The NONE isolation method described in table x12 (see 7.1.2.8) shall be applied to this command.
2h	The STRICT isolation method described in table x12 shall be applied to this command.
3h	Reserved
4h	The RANGE isolation method described in table x12 shall be applied to this command.
5h	The FUNCTIONAL isolation method described in table x12 shall be applied to this command.
6h	Reserved
7h	Vendor specific

If the ISOLATION field contains a value that the supported isolation methods attribute in the Root Information attributes page (see 7.1.2.8) indicates is not supported, 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.

…

6 Commands for OSD type devices

6.1 Summary of commands for OSD type devices

The commands for OSD type devices are listed in table 52. For the commands defined by this standard, the SERVICE ACTION field in the CDB uniquely identifies each command function. The referenced subclauses describe the service provided by each command function and the information that shall be passed to the OSD logical unit in order for it to perform that function.

Table 52 — Commands for OSD type devices

Command name	Operation code	Service action [a]	Type	Reference
…	…	…	…	…
OBJECT STRUCTURE CHECK	7Fh	8880h	M	6.j
…	…	…	…	…
READ MAP	7Fh	88B1h	M	6.j
…	…	…	…	…

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] …

6.2 APPEND

The APPEND command (see table 53) causes the specified number of bytes to be written to the designated object starting immediately after the user object's logical length.

Table 53 — APPEND command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8887h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 53.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.2.}}

6.3 CLEAR

The CLEAR command (see table 54) causes the specified number of bytes containing zero to be written to the specified user object at the specified relative location.

Table 54 — CLEAR command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8889h)							(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 54.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.3.}}

6.4 CREATE

The CREATE command (see table 55) causes the OSD device server to allocate and initialize one or more user objects.

Table 55 — CREATE command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8882h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 55.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.4.}}

6.5 CREATE AND WRITE

The CREATE AND WRITE command (see table 56) causes the OSD device server to allocate and initialize one user object and then write data to the newly created user object.

Table 56 — CREATE AND WRITE command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8892h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 56.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.5.}}

6.6 CREATE COLLECTION

The CREATE COLLECTION command (see table 57) initializes a new collection (see 4.6.6).

Table 57 — CREATE COLLECTION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8895h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 57.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.6.}}

6.7 CREATE PARTITION

The CREATE PARTITION command (see table 58) allocates and initializes a new partition.

Table 58 — CREATE PARTITION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (888Bh)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 58.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.7.}}

6.8 FLUSH

The FLUSH command (see table 59) ensures that the specified data and attribute bytes for the specified user object are written to stable storage (see 4.11).

Table 59 — 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~~			
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved		FLUSH SCOPE	
	{{The flush scope field is moved to byte 11 to make its location consistent with the basic CDB format shown in 5.2.}} {{No other changes in table 59.}}							

~~The FLUSH SCOPE field …~~

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The FLUSH SCOPE field …

{{No other changes in 6.8.}}

6.9 FLUSH COLLECTION

The FLUSH COLLECTION command (see table 61) ensures that the specified collection information and attribute bytes for the specified collection are written to stable storage (see 4.11).

Table 61 — FLUSH COLLECTION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (889Ah)							(LSB)
~~10~~	~~Reserved~~						~~FLUSH SCOPE~~	
~~11~~	~~Reserved~~		~~GET/SET CDBFMT~~		~~Reserved~~			
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved		FLUSH SCOPE	
	{{The flush scope field is moved to byte 11 to make its location consistent with the basic CDB format shown in 5.2.}} {{No other changes in table 61.}}							

~~The FLUSH SCOPE field …~~

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The FLUSH SCOPE field …

{{No other changes in 6.9.}}

6.10 FLUSH OSD

The FLUSH OSD command (see table 63) ensures that the specified data and attribute bytes for the OSD logical unit are written to stable storage (see 4.11).

Table 63 — FLUSH OSD command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (889Ch)							(LSB)
~~10~~	~~Reserved~~						~~FLUSH SCOPE~~	
~~11~~	~~Reserved~~		~~GET/SET CDBFMT~~		~~Reserved~~			
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved		FLUSH SCOPE	
	{{The flush scope field is moved to byte 11 to make its location consistent with the basic CDB format shown in 5.2.}} {{No other changes in table 63.}}							

~~The FLUSH SCOPE field …~~

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The FLUSH SCOPE field …

{{No other changes in 6.10.}}

6.11 FLUSH PARTITION

The FLUSH PARTITION command (see table 65) ensures that the specified data and attribute bytes for the specified user object are written to stable storage (see 4.11).

Table 65 — FLUSH PARTITION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (889Bh)							(LSB)
10	Reserved							FLUSH SCOPE
11	Reserved		GET/SET CDBFMT		Reserved			
~~10~~	~~Reserved~~							~~FLUSH SCOPE~~
~~11~~	~~Reserved~~		~~GET/SET CDBFMT~~		~~Reserved~~			
12	Reserved						ISOLATION	
13	Reserved		GET/SET CDBFMT		Reserved			FLUSH SCOPE
	{{The flush scope field is moved to byte 11 to make its location consistent with the basic CDB format shown in 5.2.}} {{No other changes in table 65.}}							

~~The FLUSH SCOPE field …~~

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The FLUSH SCOPE field …

{{No other changes in 6.11.}}

6.12 FORMAT OSD

The FORMAT OSD command (see table 67) causes the OSD device server to delete all user objects, delete all partitions except partition zero, and set the attributes for the root object and partition zero as defined by this standard.

Table 67 — FORMAT OSD command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8881h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 67.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

Editors Note 2 - ROW: The usefulness of isolation is dubious for FORMAT OSD because no other data transfer commands are allowed to be processed concurrently with it. However, much the same position applies for the DPO bit and FUA bit, but they are defined in the current OSD-2 revision by virtue of inclusion of the OPTIONS BYTE field.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.12.}}

6.13 GET ATTRIBUTES

The GET ATTRIBUTES command (see table 68) instructs the device server to return the specified attributes for the specified root object, partition, collection, or user object before setting the attributes, if any, specified by the command (see 4.7.4).

Table 68 — GET ATTRIBUTES command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (888Eh)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 68.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.13.}}

6.14 GET MEMBER ATTRIBUTES

The GET MEMBER ATTRIBUTES command (see table 69) instructs the device server to return the specified attributes for the specified collection and the user object members of the collection before setting the attributes, if any, specified by the command (see 4.7.4). The GET MEMBER ATTRIBUTES command is a multi-object command (see 4.6.6.2).

Table 69 — GET MEMBER ATTRIBUTES command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (88A2h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 69.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.14.}}

6.15 LIST

6.15.1 LIST command

The LIST command (see table 70) is used to obtain information from the root object or a partition.

Table 70 — LIST command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8883h)				(LSB)
~~10~~				~~Reserved~~				
10				Reserved			ISOLATION	
11	Reserved		LIST_ATTR	GET/SET CDBFMT		SORT ORDER		
				{{No other changes in table 70.}}				

The contents of the ISOLATION field are defined in 5.2.y.

The LIST_ATTR bit value combined with …

{{No other changes in 6.15.1.}}

6.16 LIST COLLECTION

The LIST COLLECTION command (see table 79) is used to get information from a collection.

Table 79 — LIST COLLECTION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8897h)				(LSB)
~~10~~				~~Reserved~~				
10				Reserved			ISOLATION	
11	Reserved		LIST_ATTR	GET/SET CDBFMT		Reserved		
				{{No other changes in table 79.}}				

The contents of the ISOLATION field are defined in 5.2.y.

The LIST_ATTR bit value combined with …

{{No other changes in 6.16.}}

6.j OBJECT STRUCTURE CHECK

{{all of 6.j is new; text markups suspended}}

6.j.1 Overview

The OBJECT STRUCTURE CHECK command (see table x5) verifies the integrity of the OSBD storage for a partition or for the root object and all partitions.

Table x5 — OBJECT STRUCTURE CHECK command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8880h)				(LSB)
10				Reserved				
11	Reserved			GET/SET CDBFMT		Reserved		
12				TIMESTAMPS CONTROL				
13								
15				Reserved				
16	(MSB)							
23				PARTITION_ID TO CHECK				(LSB)
24								
51				Reserved				
52								
79				Get and set attributes parameters (see 5.2.3)				
80								
159				Capability (see 4.9.2.2)				
160								
199				Security parameters (see 5.2.7)				

Editors Note 3 - ROW: The ISOLATION field has been omitted from the OBJECT STRUCTURE CHECK command. Prohibiting all other commands while OBJECT STRUCTURE CHECK is being processed seems like enough isolation.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The contents of the TIMESTAMPS CONTROL field are defined in 5.2.9.

The contents of the PARTITION_ID TO CHECK field specify the Partition_ID (see 4.6.4) for which the structure checking operation shall be performed. If the PARTITION_ID TO CHECK field contains zero, the structure checking shall be performed on the root object and all partitions. If the non-zero partition identified by the PARTITION_ID TO CHECK field does not exist, 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.

The get and set attributes parameters are defined in 5.2.3. 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 5.2.7.

Upon completion of an OBJECT STRUCTURE COMMAND with GOOD status:

a) The OBSD shall be ready to process application client commands to the addressed partition or to the root object without need for the processing of additional OBJECT STRUCTURE CHECK commands until a hard reset SCSI device condition is established in response to an event (see SAM-4); and
b) Any uncorrectable storage damage detected by processing the OBJECT STRUCTURE CHECK command shall be reported as described in 4.11.3.1.

The detection of uncorrectable storage damage shall not cause an OBJECT STRUCTURE CHECK command to be terminated with a CHECK CONDITION status.

Following completion of an OBJECT STRUCTURE CHECK command, the application client should send a REQUEST SENSE command to retrieve the unit attention condition, if any, established as a result of updating the damage storage reporting information as described in 4.11.3.1.

6.j.2 Structure checking for the root object and all partitions

Before modifying any OSBD storage in response to an OBJECT STRUCTURE CHECK command for the root object and all partitions, the device server shall ensure that all commands received prior to the OBJECT STRUCTURE CHECK command have completed processing.

While an OBJECT STRUCTURE CHECK command for the root object and all partitions is being processed, the root structure check accessibility attribute in the Root Always Accessible attributes page (see 7.1.2.y) shall be set to one. Upon completion of an OBJECT STRUCTURE CHECK command for the root object and all partitions, the root structure check accessibility attribute shall be set to zero.

Editors Note 4 - ROW: I believe the sense data approach (specified below) for blocking other commands while an OBJECT STRUCTURE CHECK command is being processed eliminates the need for the attributes described above and in 4.7.1. However, this opinion has yet to be confirmed by the SNIA OSD TWG.

While an OBJECT STRUCTURE CHECK command for the root object and all partitions is being processed, the device server shall terminate all commands except an INQUIRY command (see SPC-4), a REPORT LUNS command (see SPC-4), a REQUST SENSE command (see SPC-4), and a GET ATTRIBUTES command (see 6.13) that only requests attributes in the Root Always Accessible attributes page (see 7.1.2.y) or Partition Always Accessible attributes page (see 7.1.2.z) as follows:

a) a CHECK CONDITION status;
b) the sense key set to NOT READY;
c) the additional sense code set to LOGICAL UNIT NOT READY, REBUILD IN PROGRESS;
d) the INFORMATION field set to zero;
e) the SKSV bit set to one; and
f) the sense key specific data containing a progress indication as described in SPC-4.

While an OBJECT STRUCTURE CHECK command for the root object and all partitions is being processed, the device server shall complete REQUST SENSE commands with GOOD status and the following sense data:

a) the sense key set to NOT READY;
b) the additional sense code set to LOGICAL UNIT NOT READY, REBUILD IN PROGRESS;
c) the INFORMATION field set to zero;
d) the SKSV bit set to one; and
e) the sense key specific data containing a progress indication as described in SPC-4.

{{Additional sense code LOGICAL UNIT NOT READY, REBUILD IN PROGRESS is defined in SPC-4 but not identified for OSD-2 usage.}}

The processing of an OBJECT STRUCTURE CHECK command shall not affect the processing of the INQUIRY command, REPORT LUNS command, and a GET ATTRIBUTES command that only requests attributes in the Root Always Accessible attributes page or Partition Always Accessible attributes page.

6.j.3 Structure checking for a specific partition

Before modifying any OSBD storage associated with the specified partition in response to an OBJECT STRUCTURE CHECK command for a specific partition, the device server shall ensure that all commands addressed to the specified received prior to the OBJECT STRUCTURE CHECK command have completed processing.

While an OBJECT STRUCTURE CHECK command for a specific partition is being processed, the partition structure check accessibility attribute in the Partition Always Accessible attributes page (see 7.1.2.z) shall be set to one. Upon completion of an OBJECT STRUCTURE CHECK command for a specific partition, the partition structure check accessibility attribute shall be set to zero.

Editors Note 5 - ROW: I believe the sense data approach (specified below) for blocking other commands while an OBJECT STRUCTURE CHECK command is being processed eliminates the need for the attributes described above and in 4.7.1. However, this opinion has yet to be confirmed by the SNIA OSD TWG.

While an OBJECT STRUCTURE CHECK command for a specified partition is being processed, the device server shall terminate all commands addressed to that partition except a GET ATTRIBUTES command (see 6.13) that only requests attributes in the Partition Always Accessible attributes page (see 7.1.2.z) with:

a) a CHECK CONDITION status;
b) the sense key set to NOT READY;
c) the additional sense code set to LOGICAL UNIT NOT READY, REBUILD IN PROGRESS;
d) the INFORMATION field set to the Partition_ID of a partition being processed by an OBJECT STRUCTURE CHECK command;
e) the SKSV bit set to one; and
f) the sense key specific data containing a progress indication as described in SPC-4.

{{Additional sense code LOGICAL UNIT NOT READY, REBUILD IN PROGRESS is defined in SPC-4 but not identified for OSD-2 usage.}}

The processing of an OBJECT STRUCTURE CHECK command shall not affect the processing of a GET ATTRIBUTES command addressed to the partition that only requests attributes in the Partition Always Accessible attributes page.

6.17 PERFORM SCSI COMMAND

The PERFORM SCSI COMMAND command (see table 84) allows an implemented SPC-3 command (e.g., LOG SENSE) to be processed when the security method is not NOSEC (see 4.10.4). The PERFORM SCSI COMMAND command also allows an implemented SPC-4 command to be processed concurrently with attributes retrieval and setting command functions.

Table 84 — PERFORM SCSI COMMAND command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8F7Ch)							(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 84.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.17.}}

6.18 PERFORM TASK MANAGEMENT FUNCTION

The PERFORM TASK MANAGEMENT FUNCTION command (see table 86) allows a SAM-4 task management function (e.g., ABORT TASK) to be processed when the security method is not NOSEC (see 4.10.4). The PERFORM TASK MANAGEMENT FUNCTION command also allows a SAM-4 task management function to be processed concurrently with attributes retrieval and setting command functions.

Table 86 — PERFORM TASK MANAGEMENT FUNCTION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8F7Dh)							(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 86.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.18.}}

6.19 PUNCH

The PUNCH command (see table 90) removes bytes from a user object.

Table 90 — PUNCH command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8884h)				(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 90.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.19.}}

6.20 QUERY

6.20.1 Introduction

The QUERY command (see table 91) instructs the device server to return a list of the user objects that are members of the specified collection and have attributes matching the specified values. The QUERY command is a multi-object command (see 4.6.6.2).

Table 91 — QUERY command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (88A0h)				(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 91.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.20.1.}}

6.21 READ

The READ command (see table 96) requests that the device server return data to the application client from the specified user object.

Table 96 — READ command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8885h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 96.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.21.}}

6.j READ MAP

{{all of 6.j is new; text markups suspended}}

The READ command (see table x6) requests that the device server return a map of the data and attributes in the specified user object.

Table x6 — READ MAP command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (88B1h)				(LSB)
10				Reserved			ISOLATION	
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				ALLOCATION LENGTH				(LSB)
40	(MSB)							
47				DATA MAP BYTE OFFSET				(LSB)
48	(MSB)							
49				REQUESTED MAP TYPE				(LSB)
50								
51				Reserved				
52								
79				Get and set attributes parameters (see 5.2.3)				
80								
159				Capability (see 4.9.2.2)				
160								
199				Security parameters (see 5.2.7)				

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The contents of the TIMESTAMPS CONTROL field are defined in 5.2.9.

The contents of the PARTITION_ID field are defined in 5.2.6.

The contents of the USER_OBJECT_ID field are defined in 5.2.10.

The contents of the ALLOCATION LENGTH field are defined in 5.2.2.

The DATA MAP BYTE OFFSET field specifies the first byte of user data to be represented in the returned map. If the DATA MAP BYTE OFFSET 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 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.

The REQUESTED MAP TYPE field (see table x7) specifies the map descriptor type values (see table x10) that shall be returned in the parameter data.

Table x7 — REQUESTED MAP TYPE field

Code	Description
0000h	Return all map type values.
0001h	Return only WRITTEN_DATA map type values.
0002h	Return only DATA_HOLE map type values.
0003h	Return only DAMAGED_DATA map type values.
0004h to 7FFFh	Reserved
8000h	Return only DAMAGED_ATTRIBUTES map type values.
8001h to FFFFh	Reserved

The get and set attributes parameters are defined in 5.2.3. 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 4.11.2.2.

The security parameters are defined in 5.2.7.

The parameter data returned by a READ MAP command (see table x8) contains descriptors that describe the user data and attributes associated with the user object.

Table x8 — READ MAP command parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7	ADDITIONAL LENGTH (n-7)							(LSB)
	Map descriptor list							
8	Map descriptor (first)							
23								
	⋮							
n-15	Map descriptor (last)							
n								

The ADDITIONAL LENGTH field indicates the number of bytes of READ MAP command parameter data that follow. If the parameter data is truncated due to insufficient allocation length, the ADDITIONAL LENGTH field shall not be altered to reflect the truncation (i.e., the additional length indicates the number of bytes that would follow if the allocation length had been infinite). If the untruncated number of bytes that follow is greater than FFFF FFFF FFFF FFFFh the additional length shall be set to FFFF FFFF FFFF FFFFh.

Each map descriptor (see table x9) contains 16 bytes and provides information about user object attributes or one range of bytes within the user object's user data.

Table x9 — Map descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	(MSB)	MAP DESCRIPTOR TYPE						
3								(LSB)
4	(MSB)	BYTE OFFSET						
11								(LSB)
12	(MSB)	DATA LENGTH						
15								(LSB)

The MAP DESCRIPTOR TYPE field (see table x10) indicates the type of information this map descriptor contains.

Table x10 — MAP DESCRIPTOR TYPE field

Code	Name	Description
0000h		Reserved
0001h	WRITTEN_DATA	This map descriptor indicates the byte offset and data length of user data that has been written to stable storage (see 4.11) and is available for reading.
0002h	DATA_HOLE	This map descriptor indicates the byte offset and data length of a user data that lies between two WRITTEN_DATA regions, but for which no user data has been written.
0003h	DAMAGED_DATA	This map descriptor indicates the byte offset and data length of user data in which uncorrectable damage has been detected (see 4.11.3).
0004h to 7FFFh		Reserved
8000h	DAMAGED_ATTRIBUTES	This map descriptor indicates that one or more user object attributes contain uncorrectable damage.
8001h to FFFFh		Reserved

If the map descriptor type is greater than 7FFFh, the BYTE OFFSET field is reserved. If the map descriptor type is less than 8000h, the BYTE OFFSET field indicates the starting byte address of the user data that this map descriptor represents. The byte offset in the first map descriptor shall be equal to or greater than the contents of the DATA MAP

BYTE OFFSET field in the CDB. The byte offset in any map descriptor after the first shall be greater than or equal to the sum of the byte offset and data length in the preceding map descriptor.

If the map descriptor type is greater than 7FFFh, the DATA LENGTH field is reserved. If the map descriptor type is less than 8000h, the DATA LENGTH field indicates the number of bytes of user data, starting at byte offset, that this map descriptor represents.

If the READ MAP parameter data contains a map descriptor with the MAP DESCRIPTOR TYPE field set to DAMAGED_ATTRIBUTES, then that map descriptor shall be the last one in the parameter data.

6.22 REMOVE

The REMOVE command (see table 97) deletes a user object.

Table 97 — REMOVE command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (888Ah)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 97.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.22.}}

6.23 REMOVE COLLECTION

The REMOVE COLLECTION command (see table 98) removes a collection (see 4.6.6) from a partition.

Table 98 — REMOVE COLLECTION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8896h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			FCR
	{{No other changes in table 98.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

~~The FCR (force collection removal) bit …~~

~~If the FCR bit is set to one, …~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The FCR (force collection removal) bit …

If the FCR bit is set to one, …

{{No other changes in 6.23.}}

6.24 REMOVE MEMBER OBJECTS

The REMOVE MEMBER OBJECTS command (see table 99) instructs the device server to remove all the user objects that are members of the specified collection. The REMOVE MEMBER OBJECTS command is a multi-object command (see 4.6.6.2).

Table 99 — REMOVE MEMBER OBJECTS command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (88A1h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 99.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.24.}}

6.25 REMOVE PARTITION

The REMOVE PARTITION command (see table 100) deletes a partition from the OSD logical unit. If there are any collections or user objects in the partition, the command 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 PARTITION OR COLLECTION CONTAINS USER OBJECTS.

Table 100 — REMOVE PARTITION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (888Ch)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 100.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.25.}}

6.26 SET ATTRIBUTES

The SET ATTRIBUTES command (see table 101) sets the specified attributes for the specified root object, partition, collection, or user object before attributes are retrieved (see 4.7.4).

Table 101 — SET ATTRIBUTES command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (888Fh)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 101.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.26.}}

6.27 SET KEY

The SET KEY command (see table 102) causes the OSD device server to update the specified secret key.

Table 102 — SET KEY command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8898h)							(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved		KEY TO SET	
	{{No other changes in table 102.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.27.}}

6.28 SET MASTER KEY

6.28.1 Introduction

The SET MASTER KEY command (see table 104) causes the OSD device server to update the master key secret key.

Table 104 — SET MASTER KEY command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8899h)							(LSB)
~~10~~	~~Reserved~~							
10	Reserved					ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved		DH_STEP	
	{{No other changes in table 104.}}							

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

The DH_STEP (Diffie-Hellman step) field …

{{No other changes in 6.28.1.}}

6.29 SET MEMBER ATTRIBUTES

The SET MEMBER ATTRIBUTES command (see table 108) instructs the device server to set the specified attributes for the specified collection and user object members of the collection before retrieving the attributes, if any, specified by the command (see 4.7.4). The SET MEMBER ATTRIBUTES command is a multi-object command (see 4.6.6.2).

Table 108 — SET MEMBER ATTRIBUTES command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (88A3h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 108.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.29.}}

6.30 WRITE

The WRITE command (see table 109) causes the specified number of bytes to be written to the specified user object at the specified relative location.

Table 109 — WRITE command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9	SERVICE ACTION (8886h)							(LSB)
~~10~~	~~OPTIONS BYTE~~							
10	Reserved			DPO	FUA	ISOLATION		
11	Reserved		GET/SET CDBFMT		Reserved			
	{{No other changes in table 109.}}							

~~The contents of the OPTIONS BYTE field are defined in 5.2.5.~~

The contents of the DPO bit and the FUA bit are defined in 5.2.x.

The contents of the ISOLATION field are defined in 5.2.y.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.3.

{{No other changes in 6.30.}}

…

7 Parameters for OSD type devices

…

7.1.2.1 Attributes pages overview

…

Table 110 — Attributes pages defined by this standard (page 1 of 2)

Page Number	Page Name	Page Format Defined	Support Requirements	Reference
0h	User Object Directory	No	Mandatory	7.1.2.7
1h	User Object Information	No	Mandatory	7.1.2.11
2h	User Object Quotas	Yes	Mandatory	7.1.2.14
3h	User Object Timestamps	Yes	Mandatory	7.1.2.18
4h	Collections	Yes	Optional	7.1.2.19
5h	User Object Policy/Security	Yes	Mandatory	7.1.2.23
6h	User Object Error Recovery	Yes	Mandatory	7.1.2.x
~~6h~~ 7h to 7Fh	Reserved			

Table 110 — Attributes pages defined by this standard (page 2 of 2)

Page Number	Page Name	Page Format Defined	Support Requirements	Reference
C+0h	Collection Directory	No	Mandatory	7.1.2.6
C+1h	Collection Information	No	Mandatory	7.1.2.10
C+2h	Reserved			
C+3h	Collection Timestamps	Yes	Mandatory	7.1.2.17
C+4h	Reserved			
C+5h	Collection Policy/Security	Yes	Mandatory	7.1.2.22
C+6h	Collection Error Recovery	Yes	Mandatory	7.1.2.w
~~C+6h~~ C+7h to C+7Fh	Reserved			
P+0h	Partition Directory	No	Mandatory	7.1.2.5
P+1h	Partition Information	No	Mandatory	7.1.2.9
P+2h	Partition Quotas	Yes	Mandatory	7.1.2.13
P+3h	Partition Timestamps	Yes	Mandatory	7.1.2.16
P+4h	Reserved			
P+5h	Partition Policy/Security	Yes	Mandatory	7.1.2.21
P+6h	Partition Error Recovery	Yes	Mandatory	7.1.2.v
~~P+6h~~ P+7h to P+7Eh ~~P+7Fh~~	Reserved			
P+7Fh	Partition Always Accessible	Yes	Mandatory	7.1.2.z
R+0h	Root Directory	No	Mandatory	7.1.2.4
R+1h	Root Information	No	Mandatory	7.1.2.8
R+2h	Root Quotas	Yes	Mandatory	7.1.2.12
R+3h	Root Timestamps	Yes	Mandatory	7.1.2.15
R+4h	Reserved			
R+5h	Root Policy/Security	Yes	Mandatory	7.1.2.20
R+6h	Root Error Recovery	Yes	Mandatory	7.1.2.u
~~P+6h~~ P+7h to P+7Eh ~~P+7Fh~~	Reserved			
P+7Fh	Root Always Accessible	Yes	Mandatory	7.1.2.y
F000 0000h to FFFF FFFDh	Reserved			
FFFF FFFEh	Current Command	Yes	Mandatory	7.1.2.24

7.1.2.8 Root Information attributes page

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

Table 116 — Root Information attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h to 2h		Reserved	No	Yes
3h	20	OSD System ID	No	Yes
4h	8	Vendor identification	No	Yes
5h	16	Product identification	No	Yes
6h	32	Product model	No	Yes
7h	4	Product revision level	No	Yes
8h	variable	Product serial number	No	Yes
9h	variable	OSD name	Yes	No
Ah to 7Fh		Reserved	No	
80h	8	Total capacity	No	Yes
81h	8	Used capacity	No	Yes
82h		Reserved		
83h	4	Object accessibility	Yes	No
~~82h~~ 84h to BFh		Reserved	No	
C0h	8	Number of partitions	No	Yes
C1h to FFh		Reserved	No	
100h	6	Clock	No	Yes
101h to 10Fh		Reserved		
110h	1	Default isolation method	Yes	No
111h	32	Supported isolation methods	No	Yes
112h to 11Fh		Reserved		
120h	8	Data atomicity guarantee	No	Yes
121h	8	Data atomicity alignment	No	Yes
122h	8	Attributes atomicity guarantee	No	Yes
123h	1	Data/attributes atomicity multiplier	No	Yes
~~101h~~ 124h to FFFF FFFEh		Reserved	No	

…

The object accessibility attribute (83h) specifies the accessibly of the root object, all partitions, all collections, and all user objects using one of the values shown in table x11. The object accessibility attribute shall be enforced as described in 4.7.2. The object accessibility attribute in the Root Information attributes page shall be set to zero (i.e., allow all accesses) by a FORMAT OSD command.

Table x11 — Object accessibility attribute values

Code	Description
0000 0000h	Allow all accesses
0000 0001h	Deny all write accesses and allow all read accesses
0000 0000h to FFFF FFFFh	Reserved

…

The default isolation method attribute (111h) specifies the default method for isolating the actions of one command from the actions of other concurrent commands using one of the values shown in table x12. The default isolation method attribute shall be set to one (i.e., NONE) by a FORMAT OSD command.

Table x12 — Default isolation method attribute values

Code	Name	Type	Description
00h			Reserved [a]
01h	NONE	M	The actions of one command are not isolated from the actions of other concurrent commands being processed by the device server.
02h	STRICT	M	The device server shall isolate all the actions of one command from the actions of other concurrent commands (i.e., all commands are processed as if the Task Attribute (see SAM-4) is set to ORDERED).
03h			Reserved
04h	RANGE	O	The device server shall isolate the actions one command that modify a range of bytes within a user object from the actions of other concurrent commands that modify the same range of bytes within the same user object.
05h	FUNCTIONAL	O	The device server shall isolate the command function actions (see table 37 in 4.14.2.1) of one command from the same command function actions of other concurrent commands.
06h			Reserved
07h			Vendor specific
08h to FFh			Reserved [a]
Type Key: M = Command implementation is mandatory. O = Command implementation is optional.			
[a] There is no way to represent codes in this range in the ISOLATION field (see 5.2.y).			

If a set attributes list (see 5.2.3.4) attempts to set the default isolation method attribute to a value that the supported isolation methods attribute indicates is not supported, 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 5.2.3.3) attempts to set the default isolation method attribute to a value that the supported isolation methods attribute indicates is not supported, 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.

The supported isolation methods attribute (112h) is a bit mask (see table x13) that indicates which isolation methods (see table x12) are supported by the OBSD.

Table x13 — Supported isolation methods attribute contents

Bit / Byte	7	6	5	4	3	2	1	0
0	VS	Reserved	FUNC	RANGE	Reserved	STRICT	NONE	Reserved
1	Reserved							
31								

If the FUNCTIONAL isolation method (see table x12) is supported, the FUNC bit shall be set to one. If the FUNCTIONAL isolation method is not supported, the FUNC bit shall be set to zero.

If the RANGE isolation method (see table x12) is supported, the RANGE bit shall be set to one. If the RANGE isolation method is not supported, the RANGE bit shall be set to zero.

The STRICT bit shall be set to one to indicate that the STRICT isolation method (see table x12) is supported.

The NONE bit shall be set to one to indicate that the NONE isolation method (see table x12) is supported.

The data atomicity guarantee attribute (120h), the data atomicity alignment attribute (121h), the attributes atomicity guarantee attribute (122h), and the data/attributes atomicity multiplier attribute are defined in table x2 (see 4.9.2).

…

7.1.2.9 Partition Information attributes page

The Partition Information attributes page (P+1h) shall contain the attributes listed in table 117.

Table 117 — 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		Reserved		
83h	4	Object accessibility	Yes	No
~~82h~~ 84h to C0h		Reserved	No	
C1h	8	Number of collections and user objects	No	Yes
C2h to D0h		Reserved	No	
D1h	0 or 8	Actual data space	No	Yes
D2h	0 or 8	Reserved data space	Yes	No
D3h to FFFF FFFEh		Reserved	No	

…

The object accessibility attribute (83h) specifies the accessibly of the partition, all collections in the partition, and all user objects in the partition using one of the values shown in table x11 (see 7.1.2.8). The object accessibility attribute shall be enforced as described in 4.7.2. The object accessibility attribute in the Partition Information attributes page shall be set to zero (i.e., allow all accesses) by a CREATE PARTITION command.

…

7.1.2.10 Collection Information attributes page

The Collection Information attributes page (C+1h) shall contain the attributes listed in table 118.

Table 118 — 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
Dh to 80h		Reserved	No	
81h	8	Used capacity	No	Yes
82h		Reserved		
83h	4	Object accessibility	Yes	No
~~82h~~ 84h to FFFF FFFEh		Reserved	No	

…

The object accessibility attribute (83h) specifies the accessibly of the collection using one of the values shown in table x11 (see 7.1.2.8). The object accessibility attribute shall be enforced as described in 4.7.2. The object accessibility attribute in the Collection Information attributes page shall be set to zero (i.e., allow all accesses) by a CREATE COLLECTION command.

…

7.1.2.11 User Object Information attributes page

The User Object Information attributes page (1h) shall contain the attributes listed in table 120.

Table 120 — 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	4	Object accessibility	Yes	No
~~83h~~ 84h to D0h		Reserved	No	
D1h	0 or 8	Actual data space	No	Yes
D2h	0 or 8	Reserved data space	Yes	No
D3h to FFFF FFFEh		Reserved	No	

…

The object accessibility attribute (83h) specifies the accessibly of the user object using one of the values shown in table x11 (see 7.1.2.8). The object accessibility attribute shall be enforced as described in 4.7.2. The object accessibility attribute in the User Object Information attributes page shall be set to zero (i.e., allow all accesses) by a CREATE command or a CREATE AND WRITE command.

…

7.1.2.23 User Object Policy/Security attributes page

…

7.1.2.u Root Error Recovery attributes page

{{all of 7.1.2.u is new; markups suspended}}

The Root Error Recovery attributes page (R+6h) shall contain the attributes listed in table x14.

Table x14 — Root Error Recovery attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	1	Root damage summary	Yes	Yes
2h	1	Contained objects damage summary	No	Yes
3h	6	Last damaged object data time	No	Yes
4h	6	Last damaged object attributes time	No	Yes
5h	6	Last damaged contained object time	No	Yes
6h	8	Number of damaged partitions	No	Yes
7h 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 Root Error Recovery".

The root damage summary attribute (1h) indicates the overall error recovery status of the root object using the format shown in table x15.

Table x15 — Root damage summary attribute value

Bit	7	6	5	4	3	2	1	0
	P_OSC	Reserved			P_OSC_RC	R_OSC_RC	ATTR	P_LIST

If the P_OSC (partition object structure check) bit is set to zero, no partitions are processing an OBJECT STRUCTURE CHECK command (see 6.j). If the P_OSC bit is set to one, one or more partitions are processing an OBJECT STRUCTURE CHECK command.

If the P_OSC_RC (partition object structure check recommended) bit is set to zero, the processing of an OBJECT STRUCTURE CHECK command is not recommended for any partitions. If the P_OSC_RC bit is set to one, one or more partitions may benefit from the processing of an OBJECT STRUCTURE CHECK command. The partition damage summary attribute in each Partition Error Recovery attributes page (see 7.1.2.v) indicates which partitions may benefit from the processing of an OBJECT STRUCTURE CHECK command.

A P_OSC_RC bit that is set to one does not require the processing of an OBJECT STRUCTURE CHECK command on one or more partitions. When the processing of such an OBJECT STRUCTURE CHECK command is required, the process described in 4.11.3.3 is used.

If the R_OSC_RC (root object structure check recommended) bit is set to zero, the processing of an OBJECT STRUCTURE CHECK command is not recommended for the root object and all partitions. If the R_OSC_RC bit is set to one, the processing of an OBJECT STRUCTURE CHECK command is recommended for the root object and all partitions.

An R_OSC_RC bit that is set to one does not require the processing of an OBJECT STRUCTURE CHECK command for the root object and all partitions. When the processing of such an OBJECT STRUCTURE CHECK command is required, the process described in 4.11.3.3 is used.

If the ATTR (attributes) bit is set to zero, no uncorrectable damage has been detected in root object attributes. If the ATTR bit is set to one, uncorrectable damage has been detected in one or more root object attributes.

If the P_LIST (partition list) bit is set to zero, no uncorrectable damage has been detected in the list of partitions in the root object. If the P_LIST bit is set to one, uncorrectable damage has been detected in the list of partitions in the root object.

If the application client sets the root damage summary attribute to any value, the device server shall recompute the attribute's contents.

The contained objects damage summary attribute (2h) indicates the overall error recovery status of all partitions, all collections, and all user objects using the format shown in table x16.

Table x16 — Contained objects damage summary root attribute value

Bit	7	6	5	4	3	2	1	0
	Reserved						C_ATTR	C_DATA

If the c_ATTR (contained attributes) bit is set to zero, no uncorrectable damage has been detected in any attributes associated with a partition, a collection or a user object. If the c_ATTR bit is set to one, uncorrectable damage has been detected in one or more attributes associated with one or more partitions, collections or user objects.

If the C_DATA (contained data) bit is set to zero, no uncorrectable damage has been detected the contained data of any partition, collection, or user object. If the C_DATA bit is set to one, uncorrectable damage has been detected one or more of the following contained data regions:

a) The list of collections and user objects in one or more partitions;
b) The list of user objects in one or more collections; or
c) The user data contained in one or more user objects.

The last damaged object data time attribute (3h) contains the value of the clock attribute in the Root Information attributes page (see 7.1.2.8) when uncorrectable damage was most recently detected in the partition list of the root object. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field (see 5.2.9) and the bypass timestamps attribute in the Root Timestamps attributes page (see 7.1.2.15) shall not affect the updating of the last damaged object data time attribute.

The last damaged object attributes time attribute (4h) contains the value of the clock attribute in the Root Information attributes page when uncorrectable damage was most recently detected in a root object attribute. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field and the bypass timestamps attribute in the Root Timestamps attributes page shall not affect the updating of the last damaged object attributes time attribute.

The last damaged contained object time attribute (5h) contains the value of the clock attribute in the Root Information attributes page when uncorrectable damage was most recently detected in any of the following:

a) The list of collections and user objects in one or more partitions;
b) A partition attribute;
c) The list of user objects in one or more collections;
d) A collection attribute;

e) The user data contained in one or more user objects; or
f) A user object attribute.

The last damaged contained object time attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field and the bypass timestamps attribute in the Root Timestamps attributes page shall not affect the updating of the last damaged contained object time attribute.

The number of damaged partitions attribute (6h) contains the number of partitions that have unrecovered uncorrectable damage in any of the following:

a) Their list of member collections and user objects;
b) A partition attribute;
c) The list of user objects in one or more member collections;
d) A collection attribute;
e) The user data contained in one or more member user objects; or
f) A user object attribute.

If a set attributes list (see 5.2.3.4) contains an entry specifying the number of an attribute that table x14 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 5.2.3.3) specifies the number of an attribute that table x14 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.

The page format for the Root Error Recovery attributes page is shown in table x17.

Table x17 — Root Error Recovery attributes page format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE NUMBER (R+6h)				
3								(LSB)
4	(MSB)			PAGE LENGTH (1Ch)				
7								(LSB)
8	(MSB)			NUMBER OF DAMAGED PARTITIONS				
15								(LSB)
16	ROOT DAMAGE SUMMARY							
17	CONTAINED OBJECTS DAMAGE SUMMARY							
18	(MSB)			LAST DAMAGED OBJECT DATA TIME				
23								(LSB)
24	(MSB)			LAST DAMAGED OBJECT ATTRIBUTES TIME				
29								(LSB)
30	(MSB)			LAST DAMAGED CONTAINED OBJECT TIME				
35								(LSB)

The PAGE NUMBER field contains the attributes page number of the Root Error Recovery attributes page.

The PAGE LENGTH field contains the number of additional bytes in the page format of the Root Error Recovery attributes page.

The NUMBER OF DAMAGED PARTITIONS field contains the value of the number of damaged partitions attribute.

The ROOT DAMAGE SUMMARY field contains the value of the root damage summary attribute.

The CONTAINED OBJECTS DAMAGE SUMMARY field contains the value of the contained objects damage summary attribute.

The LAST DAMAGED OBJECT DATA TIME field contains the value of the last damaged object data time attribute.

The LAST DAMAGED OBJECT ATTRIBUTES TIME field contains the value of the last damaged object attributes time attribute.

The LAST DAMAGED CONTAINED OBJECT TIME field contains the value of the last damaged contained object time attribute.

7.1.2.v Partition Error Recovery attributes page

{{all of 7.1.2.v is new; text markups suspended}}

The Partition Error Recovery attributes page (P+6h) shall contain the attributes listed in table x18.

Table x18 — Partition Error Recovery attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	1	Partition damage summary	Yes	Yes
2h	1	Contained objects damage summary	No	Yes
3h	6	Last damaged object data time	No	Yes
4h	6	Last damaged object attributes time	No	Yes
5h	6	Last damaged contained object time	No	Yes
6h	8	Number of damaged objects	No	Yes
7h 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 Partition Error Recovery".

The partition damage summary attribute (1h) indicates the overall error recovery status of the partition using the format shown in table x19.

Table x19 — Partition damage summary attribute value

Bit	7	6	5	4	3	2	1	0
	Reserved				P_OSC_RC	Reserved	ATTR	M_LIST

If the P_OSC_RC (partition object structure check recommended) bit is set to zero, the processing of an OBJECT STRUCTURE CHECK command is not recommended for the partition. If the P_OSC_RC bit is set to one, the partition may benefit from the processing of an OBJECT STRUCTURE CHECK command.

A P_OSC_RC bit that is set to one does not require the processing of an OBJECT STRUCTURE CHECK command on the partition. When the processing of such an OBJECT STRUCTURE CHECK command is required, the process described in 4.11.3.3 is used.

If the ATTR (attributes) bit is set to zero, no uncorrectable damage has been detected in partition attributes. If the ATTR bit is set to one, uncorrectable damage has been detected in one or more partition attributes.

If the M_LIST (member list) bit is set to zero, no uncorrectable damage has been detected in the list of collections and user objects that are members of the partition. If the M_LIST bit is set to one, uncorrectable damage has been detected in the list of collections and user objects that are members of the partition.

If the application client sets the partition damage summary attribute to any value, the device server shall recompute the attribute's contents.

The contained objects damage summary attribute (2h) indicates the overall error recovery status of all collections and all user objects in the partition using the format shown in table x20.

Table x20 — Contained objects damage summary partition attribute value

Bit	7	6	5	4	3	2	1	0
	Reserved						C_ATTR	C_DATA

If the c_ATTR (contained attributes) bit is set to zero, no uncorrectable damage has been detected in any attributes associated with a collection or a user object in the partition. If the c_ATTR bit is set to one, uncorrectable damage has been detected in one or more attributes associated with one or more collections or user objects in the partition.

If the C_DATA (contained data) bit is set to zero, no uncorrectable damage has been detected the contained data in any collection or user object in the partition. If the C_DATA bit is set to one, uncorrectable damage has been detected one or more of the following contained data regions:

a) The list of user objects in one or more collections; or
b) The user data contained in one or more user objects.

The last damaged object data time attribute (3h) contains the value of the clock attribute in the Root Information attributes page (see 7.1.2.8) when uncorrectable damage was most recently detected in the list of collections and user objects that are members the partition. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field (see 5.2.9) and the bypass timestamps attribute in the Root Timestamps attributes page (see 7.1.2.15) shall not affect the updating of the last damaged object data time attribute.

The last damaged object attributes time attribute (4h) contains the value of the clock attribute in the Root Information attributes page when uncorrectable damage was most recently detected in a partition attribute. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field and the bypass timestamps attribute in the Root Timestamps attributes page shall not affect the updating of the last damaged object attributes time attribute.

The last damaged contained object time attribute (5h) contains the value of the clock attribute in the Root Information attributes page when uncorrectable damage was most recently detected in any of the following:

a) The list of user objects in one or more member collections;

b) A collection attribute;
c) The user data contained in one or more member user objects; or
d) A user object attribute.

The last damaged contained object time attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field and the bypass timestamps attribute in the Root Timestamps attributes page shall not affect the updating of the last damaged contained object time attribute.

The number of damaged objects attribute (6h) contains the number of member collections and user objects that have unrecovered uncorrectable damage in any of the following:

a) The list of user objects in one or more member collections;
b) A collection attribute;
c) The user data contained in one or more member user objects; or
d) A user object attribute.

If a set attributes list (see 5.2.3.4) contains an entry specifying the number of an attribute that table x18 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 5.2.3.3) specifies the number of an attribute that table x18 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.

The page format for the Partition Error Recovery attributes page is shown in table x21.

Table x21 — Partition Error Recovery attributes page format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE NUMBER (P+6h)				
3								(LSB)
4	(MSB)			PAGE LENGTH (1Ch)				
7								(LSB)
8	(MSB)			NUMBER OF DAMAGED OBJECTS				
15								(LSB)
16	PARTITION DAMAGE SUMMARY							
17	CONTAINED OBJECTS DAMAGE SUMMARY							
18	(MSB)			LAST DAMAGED OBJECT DATA TIME				
23								(LSB)
24	(MSB)			LAST DAMAGED OBJECT ATTRIBUTES TIME				
29								(LSB)
30	(MSB)			LAST DAMAGED CONTAINED OBJECT TIME				
35								(LSB)

The PAGE NUMBER field contains the attributes page number of the Partition Error Recovery attributes page.

The PAGE LENGTH field contains the number of additional bytes in the page format of the Partition Error Recovery attributes page.

The NUMBER OF DAMAGED OBJECTS field contains the value of the number of damaged objects attribute.

The PARTITION DAMAGE SUMMARY field contains the value of the partition damage summary attribute.

The CONTAINED OBJECTS DAMAGE SUMMARY field contains the value of the contained objects damage summary attribute.

The LAST DAMAGED OBJECT DATA TIME field contains the value of the last damaged object data time attribute.

The LAST DAMAGED OBJECT ATTRIBUTES TIME field contains the value of the last damaged object attributes time attribute.

The LAST DAMAGED CONTAINED OBJECT TIME field contains the value of the last damaged contained object time attribute.

7.1.2.w Collection Error Recovery attributes page

{{all of 7.1.2.w is new; text markups suspended}}

The Collection Error Recovery attributes page (C+6h) shall contain the attributes listed in table x22.

Table x22 — Collection Error Recovery attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	1	Collection damage summary	Yes	Yes
2h		Reserved	No	
3h	6	Last damaged data time	No	Yes
4h	6	Last damaged attributes time	No	Yes
5h 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 Collection Error Recovery".

The collection damage summary attribute (1h) indicates the overall error recovery status of the collection using the format shown in table x23.

Table x23 — Collection damage summary attribute value

Bit	7	6	5	4	3	2	1	0
	Reserved						ATTR	C_LIST

If the ATTR (attributes) bit is set to zero, no uncorrectable damage has been detected in collection attributes. If the ATTR bit is set to one, uncorrectable damage has been detected in one or more collection attributes.

If the C_LIST (collection list) bit is set to zero, no uncorrectable damage has been detected in the list of user objects that are members of the collection. If the C_LIST bit is set to one, uncorrectable damage has been detected in the list of user objects that are members of the collection.

If the application client sets the collection damage summary attribute to any value, the device server shall recompute the attribute's contents.

The last damaged object data time attribute (3h) contains the value of the clock attribute in the Root Information attributes page (see 7.1.2.8) when uncorrectable damage was most recently detected in the list of user objects that are members the collection. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field (see 5.2.9) and the bypass timestamps attribute in the Root Timestamps attributes page (see 7.1.2.15) shall not affect the updating of the last damaged object data time attribute.

The last damaged object attributes time attribute (4h) contains the value of the clock attribute in the Root Information attributes page when uncorrectable damage was most recently detected in a collection attribute. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field and the bypass timestamps attribute in the Root Timestamps attributes page shall not affect the updating of the last damaged object attributes time attribute.

If a set attributes list (see 5.2.3.4) contains an entry specifying the number of an attribute that table x22 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 5.2.3.3) specifies the number of an attribute that table x22 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.

The page format for the Collection Error Recovery attributes page is shown in table x24.

Table x24 — Collection Error Recovery attributes page format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE NUMBER (C+6h)				
3								(LSB)
4	(MSB)			PAGE LENGTH (Eh)				
7								(LSB)
8	COLLECTION DAMAGE SUMMARY							
9	Reserved							
10	(MSB)			LAST DAMAGED DATA TIME				
15								(LSB)
16	(MSB)			LAST DAMAGED ATTRIBUTES TIME				
21								(LSB)

The PAGE NUMBER field contains the attributes page number of the Collection Error Recovery attributes page.

The PAGE LENGTH field contains the number of additional bytes in the page format of the Collection Error Recovery attributes page.

The COLLECTION DAMAGE SUMMARY field contains the value of the collection damage summary attribute.

The LAST DAMAGED DATA TIME field contains the value of the last damaged data time attribute.

The LAST DAMAGED ATTRIBUTES TIME field contains the value of the last damaged attributes time attribute.

7.1.2.x User Object Error Recovery attributes page

{{all of 7.1.2.x is new; text markups suspended}}

The User Object Error Recovery attributes page (6h) shall contain the attributes listed in table x25.

Table x25 — User Object Error Recovery attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	1	User object damage summary	Yes	Yes
2h		Reserved	No	
3h	6	Last damaged data time	No	Yes
4h	6	Last damaged attributes time	No	Yes
5h 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 User Object Error Recovery".

The user object damage summary attribute (1h) indicates the overall error recovery status of the user object using the format shown in table x26.

Table x26 — User object damage summary attribute value

Bit	7	6	5	4	3	2	1	0
	Reserved						ATTR	DATA

If the ATTR (attributes) bit is set to zero, no uncorrectable damage has been detected in user object attributes. If the ATTR bit is set to one, uncorrectable damage has been detected in one or more user object attributes.

If the DATA bit is set to zero, no uncorrectable damage has been detected in user object's user data. If the DATA bit is set to one, uncorrectable damage has been detected in user object's user data. The READ MAP command (see 6.j) may be used to determine details of the uncorrectable damage in a user object's user data.

If the application client sets the user object damage summary attribute to any value, the device server shall recompute the attribute's contents.

The last damaged object data time attribute (3h) contains the value of the clock attribute in the Root Information attributes page (see 7.1.2.8) when uncorrectable damage was most recently detected in user object's user data. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field (see 5.2.9) and the bypass timestamps attribute in the Root Timestamps attributes page (see 7.1.2.15) shall not affect the updating of the last damaged object data time attribute.

The last damaged object attributes time attribute (4h) contains the value of the clock attribute in the Root Information attributes page when uncorrectable damage was most recently detected in a user object attribute. The attribute shall not be modified when an application client corrects the damage. The TIMESTAMPS CONTROL field and the bypass timestamps attribute in the Root Timestamps attributes page shall not affect the updating of the last damaged object attributes time attribute.

If a set attributes list (see 5.2.3.4) contains an entry specifying the number of an attribute that table x25 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 5.2.3.3) specifies the number of an attribute that table x25 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.

The page format for the User Object Error Recovery attributes page is shown in table x27.

Table x27 — User Object Error Recovery attributes page format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
3				PAGE NUMBER (C+6h)				(LSB)
4	(MSB)							
7				PAGE LENGTH (Eh)				(LSB)
8				USER OBJECT DAMAGE SUMMARY				
9				Reserved				
10	(MSB)							
15				LAST DAMAGED DATA TIME				(LSB)
16	(MSB)							
21				LAST DAMAGED ATTRIBUTES TIME				(LSB)

The PAGE NUMBER field contains the attributes page number of the Collection Error Recovery attributes page.

The PAGE LENGTH field contains the number of additional bytes in the page format of the Collection Error Recovery attributes page.

The USER OBJECT DAMAGE SUMMARY field contains the value of the user object damage summary attribute.

The LAST DAMAGED DATA TIME field contains the value of the last damaged data time attribute.

The LAST DAMAGED ATTRIBUTES TIME field contains the value of the last damaged attributes time attribute.

7.1.2.y Root Always Accessible attributes page

{{all of 7.1.2.y is new; text markups suspended}}

The Root Always Accessible attributes page (R+7Fh) shall contain the attributes listed in table x28.

Table x28 — Root Always Accessible attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	1	Root structure check accessibility	No	Yes
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 Root Always Accessible".

The root structure check accessibility attribute (number 1h) contains a coded value (see table x29) indicating the accessibility of the root object, all partitions, all collections, and all user objects based on whether an OBJECT STRUCTURE CHECK command (see 6.j) is being performed on the root object.

Table x29 — Root structure check accessibility

Code	Description
0h	Accessible
1h	Inaccessible due to active OBJECT STRUCTURE CHECK command
2h to FFh	Reserved

If a set attributes list (see 5.2.3.4) contains an entry specifying the number of an attribute that table x28 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 5.2.3.3) specifies the number of an attribute that table x28 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.

The page format for the Root Always Accessible attributes page is shown in table x30.

Table x30 — Root Always Accessible attributes page format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE NUMBER (R+7Fh)				
3								(LSB)
4	(MSB)			PAGE LENGTH (1h)				
7								(LSB)
8	ROOT STRUCTURE CHECK ACCESSIBILITY							

The PAGE NUMBER field contains the attributes page number of the Root Always Accessible attributes page.

The PAGE LENGTH field contains the number of additional bytes in the page format of the Root Always Accessible attributes page.

The ROOT STRUCTURE CHECK ACCESSIBILITY field contains the value of the root structure check accessibility attribute.

7.1.2.z Partition Always Accessible attributes page

{{all of 7.1.2.z is new; text markups suspended}}

The Partition Always Accessible attributes page (P+7Fh) shall contain the attributes listed in table x31.

Table x31 — Partition Always Accessible attributes page contents

Attribute Number	Length (bytes)	Attribute	Application Client Settable	OSD Logical Unit Provided
0h	40	Page identification	No	Yes
1h	1	Partition structure check accessibility	No	Yes
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 Partition Always Accessible".

The partition structure check accessibility attribute (number 1h) contains a coded value (see table x32) indicating the accessibility of the partition, all collections in the partition, and all user objects in the partition based on whether an OBJECT STRUCTURE CHECK command (see 6.j) is being performed on the partition object.

Table x32 — Partition structure check accessibility attribute values

Code	Description
0h	Accessible
1h	Inaccessible due to active OBJECT STRUCTURE CHECK command
2h to FFh	Reserved

If a set attributes list (see 5.2.3.4) contains an entry specifying the number of an attribute that table x31 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 5.2.3.3) specifies the number of an attribute that table x31 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.

The page format for the Partition Always Accessible attributes page is shown in table x33.

Table x33 — Partition Always Accessible attributes page format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			PAGE NUMBER (P+7Fh)				
3								(LSB)
4	(MSB)			PAGE LENGTH (1h)				
7								(LSB)
8	PARTITION STRUCTURE CHECK ACCESSIBILITY							

The PAGE NUMBER field contains the attributes page number of the Partition Always Accessible attributes page.

The PAGE LENGTH field contains the number of additional bytes in the page format of the Partition Always Accessible attributes page.

The PARTITION STRUCTURE CHECK ACCESSIBILITY field contains the value of the partition structure check accessibility attribute.

Annex B
(Informative)

Numeric order codes

B.1 Service action codes

The variable length CDB service action codes assigned by this standard are shown in table B.1.

Table B.1 — Numerical order OSD service action codes

Service Action	Command
…	…
881Dh to ~~8880h~~ 887Fh	Reserved
8880h	OBJECT STRUCTURE CHECK
8881h	FORMAT OSD
8882h	CREATE
8883h	LIST
8884h	PUNCH
…	…
88A2h	GET MEMBER ATTRIBUTES
88A3h	SET MEMBER ATTRIBUTES
88A4h to ~~8F7Bh~~ 88B0h	Reserved
88B1h	READ MAP
88B2h to 8F7Bh	Reserved
8F7Ch	PERFORM SCSI COMMAND
8F7Dh	PERFORM TASK MANAGEMENT FUNCTION
…	…

Annex C
(Informative)

Attributes defined by this standard

C.1 Attributes list

The attributes defined by this standard are shown in table C.1.

Table C.1 — Numerical order attributes defined by this standard (page 1 of 3)

Page Number	Page Name	Attribute Number	Attribute
0h	User Object Directory	0h	"INCITS T10 User Object Directory"
		1h	"INCITS T10 User Object Information"
		2h	"INCITS T10 User Object Quotas"
		3h	"INCITS T10 User Object Timestamps"
		4h	"INCITS T10 Collections"
		5h	"INCITS T10 User Object Policy/Security"
		6h	"INCITS T10 User Object Error Recovery"
1h	User Object Information	0h	Page identification
...	...	...	...
		83h	Object accessibility
...	...	...	...
6h	User Object Error Recovery	0h	Page identification
		1h	User object damage summary
		3h	Last damaged data time
		4h	Last damaged attributes time
3000 0000h	Partition Directory	3000 0000h	"INCITS T10 Partition Directory"
		3000 0001h	"INCITS T10 Partition Information"
		3000 0002h	"INCITS T10 Partition Quotas"
		3000 0003h	"INCITS T10 Partition Timestamps"
		3000 0005h	"INCITS T10 Partition Policy/Security"
		3000 0006h	"INCITS T10 Partition Error Recovery"
		3000 007Fh	"INCITS T10 Partition Always Accessible"
3000 0001h	Partition Information	0h	Page identification
...	...	...	...
		83h	Object accessibility
...	...	...	...

Table C.1 — Numerical order attributes defined by this standard (page 2 of 3)

Page Number	Page Name	Attribute Number	Attribute
3000 0006h	Partition Error Recovery	0h	Page identification
		1h	Partition damage summary
		2h	Contained objects damage summary
		3h	Last damaged object data time
		4h	Last damaged object attributes time
		5h	Last damaged contained object time
		6h	Number of damaged objects
3000 007Fh	Partition Always Accessible	0h	Page identification
		1h	Partition structure check accessibility
6000 0000h	Collection Directory	6000 0000h	"INCITS T10 Collection Directory"
		6000 0001h	"INCITS T10 Collection Information"
		6000 0003h	"INCITS T10 Collection Timestamps"
		6000 0005h	"INCITS T10 Collection Policy/Security"
		6000 0006h	"INCITS T10 Collection Error Recovery"
6000 0001h	Collection Information	0h	Page identification
...	...	...	...
		83h	Object accessibility
...	...	...	...
6000 0006h	Collection Error Recovery	0h	Page identification
		1h	Collection damage summary
		3h	Last damaged data time
		4h	Last damaged attributes time
9000 0000h	Root Directory	9000 0000h	"INCITS T10 Root Directory"
		9000 0001h	"INCITS T10 Root Information"
		9000 0002h	"INCITS T10 Root Quotas"
		9000 0003h	"INCITS T10 Root Timestamps"
		9000 0005h	"INCITS T10 Root Policy/Security"
		9000 0006h	"INCITS T10 Root Error Recovery"
		9000 007Fh	"INCITS T10 Root Always Accessible"

Table C.1 — Numerical order attributes defined by this standard (page 3 of 3)

Page Number	Page Name	Attribute Number	Attribute
9000 0001h	Root Information	0h	Page identification
		3h	OSD System ID
		4h	Vendor identification
		5h	Product identification
		6h	Product model
		7h	Product revision level
		8h	Product serial number
		9h	OSD name
		80h	Total capacity
		81h	Used capacity
		83h	Object accessibility
		C0h	Number of partitions
		100h	Clock
		110h	Default isolation method
		111h	Supported isolation methods
		120h	Data atomicity guarantee
		121h	Data atomicity alignment
		122h	Attributes atomicity guarantee
		123h	Data/attributes atomicity multiplier
9000 0002h	Root Quotas	0h	Page identification
...	...	...	...
9000 0006h	Root Error Recovery	0h	Page identification
		1h	Root damage summary
		2h	Contained objects damage summary
		3h	Last damaged object data time
		4h	Last damaged object attributes time
		5h	Last damaged contained object time
		6h	Number of damaged partitions
9000 007Fh	Root Always Accessible	0h	Page identification
		1h	Root structure check accessibility
FFFF FFFEh	Current Command	0h	Page identification
...	...	...	...