ENDL
TEXAS

Date: 10 September 2005
To: T10 Technical Committee & SNIA OSD TWG
From: Ralph O. Weber
Subject: OSD-2 Multi-Object LIST and LIST COLLECTION command enhancements

Although collections were introduced in OSD [ANSI INCITS 400-2004] for fast indexing and multi-object operations, only a small portion of their functionality were defined in that standard. No multi-object operations were defined and fast indexing was defined based on Object ID only. This proposal specifies OSD-2 [T10/1731-D] enhancements in regards to collections.

Some of the OSD-2 features introduced in this proposal are applicable not only to collections but also partitions. This proposal describes multi-object enhancements to the LIST and LIST COLLECTION commands. Document T10/05-328 proposes the addition of the QUERY command to return the attributes from multiple objects.

This proposal is based on the OSDv2 Collections document prepared by the SNIA OSD Technical Working Group. Review inputs from the SNIA OSD TWG will be incorporated in new revisions of this proposal as they become available.

Revision History

r0 Initial proposal, based as nearly as possible on the SNIA OSD TWG OSDv2 Collections document
r1 Synchronize permissions bits definitions with other multi-object commands proposed in T10/05-328

The document of reference for all clause, subclause, table, and page numbers is osd2r00.pdf. Text shown in green is introductory or explanatory and not included in the proposed OSD-2 changes.

Changes Proposed in OSD-2

Modify the 2.2 heading as follows:

2.2 Approved ISO and ANSI references

Add the following to the end of 2.2 (note editing an ISO OSD is not funded and has not been requested in T10):

ANSI INCITS 400-2004, Object-based Storage Device Commands (OSD)

Modify 3.2 (Acronyms) as follows:

OSD Object-based Storage Device Commands (~~this standard,~~ see 2.2 ~~clause 1~~)
OSD-2 Object-based Storage Device Commands -2 (this standard, see clause 1)

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

Modify table 10 as follows:

Table 10 — Permissions bit mask format

Bit / Byte	7	6	5	4	3	2	1	0
49 ~~31~~	READ	WRITE	GET_ATTR	SET_ATTR	CREATE	REMOVE	OBJ_MGMT	APPEND
50 ~~32~~	DEV_MGMT	GLOBAL	POL/SEC	M_OBJECT	QUERY	Reserved		
51 ~~33~~	Reserved							
52 ~~34~~	Reserved							
53 ~~35~~	Reserved							

…

A multiple objects (M_OBJECT) bit set to one in combination with other permissions bits allows retrieving attributes multiple user objects, setting attributes in multiple user objects, and removing multiple user objects. An M_OBJECT bit set to zero prohibits multiple user object commands.

A QUERY bit set to one allows searching the user objects in a collection for specified attribute values. An QUERY bit set to zero prohibits searching the user objects in a collection.

Note: most uses of the query bit appear in document T10/05-328, but the next changes also reference it.

Modify the introduction to table 12 (User object/collection descriptor format) as follows:

4.9.2.2.2 U/C capability object descriptor

If the object descriptor type is U/C (i.e., 1h), the OBJECT DESCRIPTOR field shall have the format shown in table 12, specifying a single collection or user object to which the capability allows access. If the M_OBJECT permission bit is set to one or the QUERY permission bit is set to one (see 4.9.2.2.1), the U/C capability object descriptor allows access to a single collection and the attributes associated with each user object in the collection.

Modify the introduction to table 14 (Partition descriptor format) as follows:

4.9.2.2.3 PAR capability object descriptor

If the object descriptor type is PAR (i.e., 2h), the OBJECT DESCRIPTOR field shall have the format shown in table 14, specifying a single partition to which the capability allows access. For a LIST COLLECTION command with the M_OBJECT bit set to one, the PAR capability object descriptor allows access to a single partition and the attributes associated with each collection in the partition. For the LIST command with the M_OBJECT bit set to one, the PAR capability object descriptor allows access to:

a) The root object and the attributes associated with each partition; or
b) A partition and the attributes associated with each user object in the partition.

Modify table 15 as follows:

Table 15 — Commands allowed by specific capability field values

Commands allowed and CDB fields whose contents are restricted by capability field contents, if any	Capability Field values that allow a command		
	Object Type Name	Permission Bits That Are Set To One	Object Descriptor Name
⋮	⋮	⋮	⋮
A LIST command addressed to a partition with the LIST_ATTR bit to be set to zero	PARTITION	READ	PAR
A LIST command addressed to a partition	PARTITION	READ and M_OBJECT	PAR
A LIST command addressed to the root object with the LIST_ATTR bit to be set to zero	ROOT	READ	PAR
A LIST command addressed to the root object	ROOT	READ and M_OBJECT	PAR
A LIST COLLECTION command addressed to a collection with the LIST_ATTR bit to be set to zero	COLLECTION	READ	U/C
A LIST COLLECTION command addressed to a collection	COLLECTION	READ and M_OBJECT	U/C
A LIST COLLECTION command addressed to a partition with the LIST_ATTR bit to be set to zero	PARTITION	READ	PAR
A LIST COLLECTION command addressed to a partition	PARTITION	READ and M_OBJECT	PAR
⋮	⋮	⋮	⋮
Combinations of OBJECT TYPE field, PERMISSION BITS field, and OBJECT DESCRIPTOR TYPE field values not shown in this table and table 16 are reserved. The capability fields not shown in this table may place additional limits on the objects that are allowed to be accessed.			

Modify table 16 as follows:

Table 16 — Attribute retrieving and setting function allowed by specific capability field values (Sheet 1 of 2)

Attribute-Related Functions Allowed	Capability Field values that allow attribute-related functions		
	Object Type Name	Permission Bits That Are Set To One	Object Descriptor Name
Retrieval of attributes from the Current Command attributes page (see 7.1.2.24)	USER or COLLECTION	GET_ATTR	U/C
Retrieval of attributes from the Current Command attributes page	PARTITION or ROOT	GET_ATTR	PAR
Retrieval of attributes from any attributes page associated with the addressed user object	USER	GET_ATTR	U/C
As part of a CREATE command or CREATE AND WRITE command, the retrieval of attributes from any attributes page associated with any user object created by the command	USER	GET_ATTR	U/C
As part of a LIST COLLECTION command with the LIST_ATTR bit to be set to one, the return in the parameter data of attributes from any attributes page associated with each user object in the collection	COLLECTION	GET_ATTR and M_OBJECT	U/C
Retrieval of attributes from any attributes page associated with the addressed collection	COLLECTION	GET_ATTR	U/C
As part of a CREATE COLLECTION command, the retrieval of attributes from any attributes page associated with the collection created by the command	COLLECTION	GET_ATTR	U/C
As part of a LIST COLLECTION command with the LIST_ATTR bit to be set to one, the return in the parameter data of attributes from any attributes page associated with each collection in the partition	PARTITION	GET_ATTR and M_OBJECT	PAR
As part of a LIST command with the LIST_ATTR bit to be set to one, the return in the parameter data of attributes from any attributes page associated with each user object in the partition	PARTITION	GET_ATTR and M_OBJECT	PAR
Retrieval of attributes from any attributes page associated with the addressed partition	PARTITION	GET_ATTR	PAR
As part of a CREATE PARTITION command, the retrieval of attributes from any attributes page associated with the created partition	PARTITION	GET_ATTR	PAR
As part of a LIST command with the LIST_ATTR bit to be set to one, the return in the parameter data of attributes from any attributes page associated with each partition	ROOT	GET_ATTR and M_OBJECT	PAR
Combinations of OBJECT TYPE field, PERMISSION BITS field, and OBJECT DESCRIPTOR TYPE field values not shown in this table and table 15 are reserved. The capability fields not shown in this table may place additional limits on the objects that are allowed to be accessed.			

Table 16 — Attribute retrieving and setting function allowed by specific capability field values (Sheet 2 of 2)

Attribute-Related Functions Allowed	Capability Field values that allow attribute-related functions		
	Object Type Name	**Permission Bits That Are Set To One**	**Object Descriptor Name**
Retrieval of attributes from any attributes page associated with the root object or in any attributes page associated with partition zero (see 3.1.32)	ROOT	GET_ATTR	PAR
⋮	⋮	⋮	⋮
Combinations of OBJECT TYPE field, PERMISSION BITS field, and OBJECT DESCRIPTOR TYPE field values not shown in this table and table 15 are reserved. The capability fields not shown in this table may place additional limits on the objects that are allowed to be accessed.			

Modify 6.13 and 6.14 as follows:

6.13 LIST

6.13.1 LIST command

The LIST command is used to obtain information from the root object or a partition.

Table 64 — LIST command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8803h)				(LSB)
10				Reserved				
11	Reserved	LIST_ATTR		GET/SET CDBFMT		SORT ORDER		
12				TIMESTAMPS CONTROL				
13								
15				Reserved				
16	(MSB)							
23				PARTITION_ID				(LSB)
24								
31				Reserved				
	(MSB)							:::
				OBJECT_SESSION_ID				(LSB)
32	(MSB)							
35				LIST IDENTIFIER				(LSB)
36	(MSB)							
43				ALLOCATION LENGTH				(LSB)
44	(MSB)							
51				INITIAL OBJECT_ID				(LSB)
52								
79				Get and set attributes parameters (see 5.2.2)				
80								
159				Capability (see 4.9.2.2)				
160								
199				Security parameters (see 5.2.6)				

The LIST_ATTR bit value combined with the value in the PARTITION_ID field specify the information that shall be returned (see table 64a).

Table 64a — Specifying objects and attributes to be listed

PARTITION_ID field	LIST_ATTR bit	Description
zero	zero	The Partition_IDs (see 4.6.4) in the root object shall be returned in the parameter data.
	one	The Partition_IDs in the root object and attributes specified by the get and set attributes parameters (see 5.2.2) for each partition shall be returned in the parameter data.
one	zero	The User_Object_IDs in the specified partition shall be returned in the parameter data.
	one	The User_Object_IDs in the specified partition and attributes specified by the get and set attributes parameters for each user object shall be returned in the parameter data.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2. If the LIST_ATTR bit is set to one, page format attribute processing is illegal. If the LIST_ATTR bit is set to one and the GET/SET CDBFMT field contains a value other than 11b, the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

The SORT ORDER field (see table 65) specifies the order in which the returned Partition_IDs or User_Object_IDs shall be sorted.

Table 65 — LIST command sort order values

Sort Order	Description
0h	Ascending numeric value
1h to Fh	Reserved

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

The contents of the PARTITION_ID field are defined in 5.2.5. The contents of the PARTITION_ID field combined with the LIST_ATTR bit value specify the information that shall be returned (see table 64a). ~~If the PARTITION_ID field contains zero, the Partition_IDs (see 4.6.4) in the root object shall be returned. If the PARTITION_ID field contains a non-zero value, the User_Object_IDs in the specified partition shall be returned.~~

The LIST IDENTIFIER field contains zero if the INITIAL OBJECT_ID field contains Partition_ID or User_Object_ID (see 4.6.2). Otherwise, the LIST IDENTIFIER field contains the list identifier returned by a previous LIST command.

The ALLOCATION LENGTH field specifies the maximum number of bytes that an application client has allocated for the returned list. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The allocation length is used to limit the maximum amount of the list returned to an application client. The device server shall terminate transfers to the Data-In Buffer if the number of bytes specified by the ALLOCATION LENGTH field have been transferred or if all available data have been transferred, whichever is less. If the information being transferred is truncated, the contents of the ADDITIONAL LENGTH field (see table 66) shall not be altered to reflect the truncation.

The contents of the INITIAL OBJECT_ID field depend on the contents of the LIST IDENTIFIER field. If the LIST IDENTIFIER field contains zero, the INITIAL OBJECT_ID field specifies the lowest valued Partition_ID or User_Object_ID to be returned. If the LIST IDENTIFIER field contains any value other than zero, the INITIAL OBJECT_ID field contains the value in the CONTINUATION OBJECT_ID field from the same returned parameter data that contained the value in the LIST IDENTIFIER field.

~~The~~ If the LIST_ATTR bit is set to zero, the get ~~and set~~ attributes parameters are defined in 5.2.2. ~~The format of the Data-In Buffer and Data-Out Buffer when attributes are being retrieved or set is described in 4.12.~~ If the LIST_ATTR bit is set to one, the get attributes parameters are defined in 5.2.2.3 with the additional processing requirements that depend on PARTITION_ID field contents and attribute page numbers as shown in table 65a.

Table 65a — Attributes processing requirements for LIST commands with the LIST_ATTR bit set to one

PARTITION_ID field	Attribute page number values	Description
zero	R+0h to R+2FFF FFFFh	The retrieved attribute values shall be returned in the retrieved attributes segment of the Data-In Buffer (see 4.12.3) in the format defined by 7.1.3.
	P+0h to P+2FFF FFFFh	The retrieved attribute values shall be returned in the LIST command parameter data as defined in 6.13.2.
	F000 0000h to FFFF FFFFh	a) If the object associated with the attribute is the root object, the attribute value shall be returned in the retrieved attributes segment of the Data-In Buffer; or b) If the object associated with the attribute is a partition, the attribute value shall be returned in the LIST command parameter data.
	0h to 2FFF FFFFh and C+0h to C+2FFF FFFFh	The command to be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
not zero	P+0h to P+2FFF FFFFh	The retrieved attribute values shall be returned in the retrieved attributes segment of the Data-In Buffer in the format defined by 7.1.3.
	0h to 2FFF FFFFh	The retrieved attribute values shall be returned in the LIST command parameter data as defined in 6.13.2.
	F000 0000h to FFFF FFFFh	a) If the object associated with the attribute is the partition specified in the LIST command, the attribute value shall be returned in the retrieved attributes segment of the Data-In Buffer; or b) If the object associated with the attribute is a user object, the attribute value shall be returned in the LIST command parameter data.
	R+0h to R+2FFF FFFFh and C+0h to C+2FFF FFFFh	The command to be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Regardless of LIST_ATTR bit value, the set attributes parameters are defined in 5.2.2.

The format of the Data-Out Buffer when attributes are being retrieved or set is described in 4.12.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

6.13.2 LIST command parameter data

The parameter data returned by the LIST command (see table 66) contains the requested list of partitions or user objects.

Table 66 — LIST command parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7				ADDITIONAL LENGTH (n-7)				(LSB)
8	(MSB)							
15				CONTINUATION OBJECT_ID				(LSB)
16	(MSB)							
19				LIST IDENTIFIER				(LSB)
20								
22				Reserved				
23				OBJECT DESCRIPTOR FORMAT			LSTCHG	~~ROOT~~ Obsolete
~~24~~								
~~n~~				~~List of User_Object_IDs or Partition_IDs~~				
				Object descriptor list				
24								
				Object descriptor (first)				
				⋮				
n				Object descriptor (last)				

The ADDITIONAL LENGTH field indicates the number of bytes of LIST 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.

The CONTINUATION OBJECT_ID field provides information that may be used to continue a truncated list with a new LIST command. If the CONTINUATION OBJECT_ID field contains zero, the parameter data contains all of the list results and no further LIST commands are needed. If a new LIST command is sent to continue a truncated list, the contents of the CONTINUATION OBJECT_ID field are copied to the INITIAL OBJECT_ID field of that new command.

The LIST IDENTIFIER field contains an identifier required for continuing a truncated list in a new LIST command. If a new LIST command is sent to continue a truncated list, the contents of the LIST IDENTIFIER field are copied to the LIST IDENTIFIER field of that new command.

The OBJECT DESCRIPTOR FORMAT field (see table 66a) indicates the format of the object descriptors.

Table 66a — LIST command OBJECT DESCRIPTOR FORMAT field values

Code	Description	Reference
00h	The LIST command parameter data format shall be as specified in OSD (see 2.2).	
01h	The object descriptors are a list of Partition_IDs each of which is eight bytes and has the format shown in table 66b.	6.13.3.1
02h	Each object descriptor (see table 66c) contains a Partition_ID followed by attribute parameters associated with the indicated partition.	6.13.3.2
03h to 20h	Reserved	
21h	The object descriptors are a list of User_Object_IDs each of which is eight bytes and has the format shown in table 66f.	6.13.3.5
22h	Each object descriptor (see table 66g) contains a User_Object_ID followed by attribute parameters associated with the indicated user object.	6.13.3.6
23h to 3Fh	Reserved	

A LSTCHG (list has changed) bit set to zero indicates that the entries in the list of OSD objects in the parameter data has not changed since the first LIST command identified by the list identifier. A LSTCHG bit set to one indicates that the entries in the list of OSD objects in the parameter data has changed since the first LIST command identified by the list identifier and that starting the list over at the original initial object_id may be necessary in order to obtain a complete list.

~~A ROOT bit set to zero indicates that the OSD object IDs in the parameter data are from a partition and are User_Object_IDs. A ROOT set to one indicates that the OSD object IDs in the parameter data are from the root object and are Partition_IDs.~~

The parameter list ~~of User_Object_IDs or Partition_IDs contains~~ shall contain one ~~entry~~ object descriptor for each user object or partition identified by the LIST command. If the list is truncated based on allocation length, the truncation shall not occur in the middle of an object descriptor ~~a User_Object_ID or Partition_ID~~.

The ~~list~~ LIST command parameter data shall not contain Collection_Object_IDs. Lists of Collection_Object_IDs may be obtained using the LIST COLLECTION command (see 6.14).

6.13.3 LIST command and LIST COLLECTION command object descriptor formats

6.13.3.1 Partition_ID only object descriptor format

For a LIST command with the PARTITION_ID field set to zero and the LIST_ATTR bit set to zero, each parameter data object descriptor shall be eight bytes in length and shall have the format shown in table 66b.

Table 66b — Partition_ID only object descriptor format

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

The PARTITION_ID field indicates the partition (see 4.6.4) to which the object descriptor applies.

6.13.3.2 Partition_ID with partition attributes object descriptor format

For a LIST command with PARTITION_ID field set to zero and the LIST_ATTR bit set to one, each parameter data object descriptor shall have the format shown in table 66c.

Table 66c — Partition_ID with partition attributes object descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7				PARTITION_ID				(LSB)
8								
9				Reserved				
10	(MSB)							
11				ATTRIBUTES LIST LENGTH (n-11)				(LSB)
	Attributes list entries							
12								
				Attributes list entry 0 (see 7.1.3.3)				
				⋮				
n				Attributes list entry x (see 7.1.3.3)				

The PARTITION_ID field indicates the partition (see 4.6.4) to which the object descriptor applies.

The ATTRIBUTES LIST LENGTH field indicates the number of bytes of attributes list entries that follow in the object descriptor. If the parameter data is truncated due to insufficient allocation length, the ATTRIBUTES LIST 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).

Each attributes list entry shall have the format shown in 7.1.3.3 and contain information about one attribute with an attribute page number between:

a) P+0h and P+2FFF FFFFh inclusive; or
b) F000 0000h and FFFF FFFEh inclusive.

6.13.3.3 Collection_Object_ID only object descriptor format

For a LIST COLLECTION command with the COLLECTION_OBJECT_ID field set to zero and the LIST_ATTR bit set to zero, each parameter data object descriptor shall be eight bytes in length and shall have the format shown in table 66d.

Table 66d — Collection_Object_ID only object descriptor format

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

The COLLECTION_OBJECT_ID field indicates the collection (see 4.6.6) to which the object descriptor applies.

6.13.3.4 Collection_Object_ID with collection attributes object descriptor format

For a LIST COLLECTION command with COLLECTION_OBJECT_ID field set to zero and the LIST_ATTR bit set to one, each parameter data object descriptor shall have the format shown in table 66e.

Table 66e — Collection_Object_ID with collection attributes object descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7				COLLECTION_OBJECT_ID				(LSB)
8								
9				Reserved				
10	(MSB)							
11				ATTRIBUTES LIST LENGTH (n-11)				(LSB)
	Attributes list entries							
12								
				Attributes list entry 0 (see 7.1.3.3)				
				⋮				
n				Attributes list entry x (see 7.1.3.3)				

The COLLECTION_OBJECT_ID field indicates the collection (see 4.6.6) to which the object descriptor applies.

The ATTRIBUTES LIST LENGTH field indicates the number of bytes of attributes list entries that follow in the object descriptor. If the parameter data is truncated due to insufficient allocation length, the ATTRIBUTES LIST 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).

Each attributes list entry shall have the format shown in 7.1.3.3 and contains information about one attribute with an attribute page number between:

a) C+0h and C+2FFF FFFFh inclusive; or
b) F000 0000h and FFFF FFFEh inclusive.

6.13.3.5 User_Object_ID only object descriptor format

Each parameter data object descriptor shall be eight bytes in length and shall have the format shown in table 66f if the LIST_ATTR bit is set to zero for:

a) A LIST command with the PARTITION_ID field set to a value other than zero; or
b) A LIST COLLECTION command with the COLLECTION_OBJECT_ID field set to a value other than zero.

Table 66f — User_Object_ID only object descriptor format

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

The USER_OBJECT_ID field indicates the user object (see 4.6.5) to which the object descriptor applies.

6.13.3.6 User_Object_ID with user object attributes object descriptor format

Each parameter data object descriptor shall have the format shown in table 66g if the LIST_ATTR bit is set to one for:

a) A LIST command with the PARTITION_ID field set to a value other than zero; or
b) A LIST COLLECTION command with the COLLECTION_OBJECT_ID field set to a value other than zero.

Table 66g — User_Object_ID with user object attributes object descriptor format

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
7				USER_OBJECT_ID				(LSB)
8								
9				Reserved				
10	(MSB)							
11				ATTRIBUTES LIST LENGTH (n-11)				(LSB)
	Attributes list entries							
12								
				Attributes list entry 0 (see 7.1.3.3)				
				⋮				
n				Attributes list entry x (see 7.1.3.3)				

The USER_OBJECT_ID field indicates the user object (see 4.6.5) to which the object descriptor applies.

The ATTRIBUTES LIST LENGTH field indicates the number of bytes of attributes list entries that follow in the object descriptor. If the parameter data is truncated due to insufficient allocation length, the ATTRIBUTES LIST 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).

Each attributes list entry shall have the format shown in 7.1.3.3 and contain information about one attribute with an attribute page number between:

a) 0h and 2FFF FFFFh inclusive; or
b) F000 0000h and FFFF FFFEh inclusive.

6.14 LIST COLLECTION

The LIST COLLECTION command (see table 67) is used to get information from a collection.

Table 67 — LIST COLLECTION command

Bit / Byte	7	6	5	4	3	2	1	0
8	(MSB)							
9				SERVICE ACTION (8817h)				(LSB)
10	Reserved							
11	Reserved	LIST_ATTR	GET/SET CDBFMT		Reserved			
12	TIMESTAMPS CONTROL							
13				Reserved				
15								
16	(MSB)							
23				PARTITION_ID				(LSB)
24	(MSB)							
31				COLLECTION_OBJECT_ID				(LSB)
32	(MSB)							
35				LIST IDENTIFIER				(LSB)
36	(MSB)							
43				ALLOCATION LENGTH				(LSB)
44	(MSB)							
51				INITIAL OBJECT_ID				(LSB)
52								
79				Get and set attributes parameters (see 5.2.2)				
80								
159				Capability (see 4.9.2.2)				
160								
199				Security parameters (see 5.2.6)				

The LIST_ATTR bit value combined with the value in the COLLECTION_OBJECT_ID field specify the information that shall be returned (see table 67a).

Table 67a — Specifying collections and attributes to be listed

COLLECTION_ OBJECT_ID field	LIST_ATTR bit	Description
zero	zero	The Collection_Object_IDs (see 4.6.6) in the specified partition shall be returned in the parameter data.
	one	The Collection_Object_IDs in the specified partition and attributes specified by the get and set attributes parameters (see 5.2.2) for each collection shall be returned in the parameter data.
one	zero	The User_Object_IDs in the specified collection shall be returned in the parameter data.
	one	The User_Object_IDs in the specified collection and attributes specified by the get and set attributes parameters for each user object shall be returned in the parameter data.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2. If the LIST_ATTR bit is set to one, page format attribute processing is illegal. If the LIST_ATTR bit is set to one and the GET/SET CDBFMT field contains a value other than 11b, the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

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

The contents of the PARTITION_ID field are defined in 5.2.5.

The COLLECTION_OBJECT_ID field specifies Collection_Object_ID (see 4.6.6) to be processed. ~~for which a list of member User_Object_IDs shall be returned.~~ The contents of the COLLECTION_OBJECT_ID field combined with the LIST_ATTR bit value specify the information that shall be returned (see table 67a). ~~If the COLLECTION_OBJECT_ID field contains zero, the Collection_Object_IDs of all collections in the partition shall be returned.~~ If the collection identified by ~~the~~ a non-zero COLLECTION_OBJECT_ID 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 LIST IDENTIFIER field contains zero if the INITIAL OBJECT_ID field contains Collection_Object_ID or User_Object_ID (see 4.6.5). Otherwise, the LIST IDENTIFIER field contains the list identifier returned by a previous LIST COLLECTION command.

The ALLOCATION LENGTH field specifies the maximum number of bytes that an application client has allocated for the returned list. An allocation length of zero indicates that no data shall be transferred. This condition shall not be considered as an error.

The allocation length is used to limit the maximum amount of the list returned to an application client. The device server shall terminate transfers to the Data-In Buffer if the number of bytes specified by the ALLOCATION LENGTH field have been transferred or if all available data have been transferred, whichever is less. If the information being transferred is truncated, the contents of the ADDITIONAL LENGTH field (see table 68) shall not be altered to reflect the truncation.

The contents of the INITIAL OBJECT_ID field depend on the contents of the LIST IDENTIFIER field. If the LIST IDENTIFIER field contains zero, the INITIAL OBJECT_ID field specifies the lowest valued Collection_Object_ID or User_Object_ID to be returned. If the LIST IDENTIFIER field contains any value other than zero, the INITIAL OBJECT_ID field contains the

value in the CONTINUATION OBJECT_ID field from the same returned parameter data that contained the value in the LIST IDENTIFIER field.

~~The~~ If the LIST_ATTR bit is set to zero, the get ~~and set~~ attributes parameters are defined in 5.2.2. ~~The format of the Data-In Buffer and Data-Out Buffer when attributes are being retrieved or set is described in 4.12.~~ If the LIST_ATTR bit is set to one, the get attributes parameters are defined in 5.2.2.3 with the additional processing requirements that depend on COLLECTION_OBJECT_ID field contents and attribute page numbers as shown in table 67b.

Table 67b — Attributes processing requirements for LIST COLLECTION commands with the LIST_ATTR bit set to one

COLLECTION_ OBJECT_ID field	Attribute page number values	Description
zero	P+0h to P+2FFF FFFFh	The retrieved attribute values shall be returned in the retrieved attributes segment of the Data-In Buffer (see 4.12.3) in the format defined by 7.1.3.
	C+0h to C+2FFF FFFFh	The retrieved attribute values shall be returned in the LIST command parameter data as defined in 6.13.2.
	F000 0000h to FFFF FFFFh	a) If the object associated with the attribute is the partition, the attribute value shall be returned in the retrieved attributes segment of the Data-In Buffer; or b) If the object associated with the attribute is a collection, the attribute value shall be returned in the LIST command parameter data.
	0h to 2FFF FFFFh and R+0h to R+2FFF FFFFh	The command to be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.
not zero	C+0h to C+2FFF FFFFh	The retrieved attribute values shall be returned in the retrieved attributes segment of the Data-In Buffer in the format defined by 7.1.3.
	0h to 2FFF FFFFh	The retrieved attribute values shall be returned in the LIST command parameter data as defined in 6.13.2.
	F000 0000h to FFFF FFFFh	a) If the object associated with the attribute is the collection specified in the LIST COLLECTION command, the attribute value shall be returned in the retrieved attributes segment of the Data-In Buffer; or b) If the object associated with the attribute is a user object, the attribute value shall be returned in the LIST command parameter data.
	R+0h to R+2FFF FFFFh and P+0h to P+2FFF FFFFh	The command to be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

Regardless of LIST_ATTR bit value, the set attributes parameters are defined in 5.2.2.

The format of the Data-Out Buffer when attributes are being retrieved or set is described in 4.12.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

The parameter data returned by the LIST COLLECTION command (see table 68) contains the requested information about the collections in the specified partition or user objects in the specified collection.

Table 68 — LIST COLLECTION command parameter data

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			ADDITIONAL LENGTH (n-7)				
7								(LSB)
8	(MSB)			CONTINUATION OBJECT_ID				
15								(LSB)
16	(MSB)			LIST IDENTIFIER				
19								(LSB)
20				Reserved				
22								
23	OBJECT DESCRIPTOR FORMAT						LSTCHG	~~COLTN~~ Obsolete
~~24~~				~~List of User_Object_IDs or Collection_Object_IDs~~				
~~n~~								
	Object descriptor list							
24				Object descriptor (first)				
	⋮							
				Object descriptor (last)				
n								

The ADDITIONAL LENGTH field indicates the number of bytes of LIST COLLECTION 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.

The CONTINUATION OBJECT_ID field provides information that may be used to continue a truncated list with a new LIST COLLECTION command. If the CONTINUATION OBJECT_ID field contains zero, the parameter data contains all of the list results and no further LIST COLLECTION commands are needed. If a new LIST COLLECTION command is sent to continue a truncated list, the contents of the CONTINUATION OBJECT_ID field are copied to the INITIAL OBJECT_ID field of that new command.

The LIST IDENTIFIER field contains an identifier required for continuing a truncated list in a new LIST COLLECTION command. If a new LIST COLLECTION command is sent to continue a truncated list, the contents of the LIST IDENTIFIER field are copied to the LIST IDENTIFIER field of that new command.

The OBJECT DESCRIPTOR FORMAT field (see table 68a) indicates the format of the object descriptors.

Table 68a — LIST COLLECTION command OBJECT DESCRIPTOR FORMAT field values

Code	Description	Reference
00h	The LIST parameter data format shall be as specified in OSD (see 2.2).	
01h to 10h	Reserved	
11h	The object descriptors are a list of Collection_Object_IDs each of which is eight bytes and has the format shown in table 66d.	6.13.3.3
12h	Each object descriptor (see table 66e) contains a Collection_Object_ID followed by attribute parameters associated with the indicated collection.	6.13.3.4
13h to 20h	Reserved	
21h	The object descriptors are a list of User_Object_IDs each of which is eight bytes and has the format shown in table 66f.	6.13.3.5
22h	Each object descriptor (see table 66g) contains a User_Object_ID followed by attribute parameters associated with the indicated user object.	6.13.3.6
23h to 3Fh	Reserved	

A LSTCHG (list has changed) bit set to zero indicates that the entries in the list of OSD objects in the parameter data has not changed since the first LIST COLLECTION command identified by the list identifier. A LSTCHG bit set to one indicates that the entries in the list of OSD objects in the parameter data has changed since the first LIST COLLECTION command identified by the list identifier and that starting the list over at the original initial object_id may be necessary in order to obtain a complete list.

~~A COLTN bit of zero indicates that the OSD object IDs in the parameter data are from a collection and are User_Object_IDs. A COLTN bit of one indicates that the OSD object IDs in the parameter data are from the partition and are Collection_Object_IDs.~~

The parameter list ~~of User_Object_IDs or Collection_Object_IDs contains~~ shall contain one ~~entry~~ object descriptor for each user object or collection identified by the LIST COLLECTION command. If the list is truncated based on allocation length, the truncation shall not occur in the middle of an object descriptor ~~a User_Object_ID or Collection_Object_ID~~.

Modify 7.1.3 as follows:

7.1.3 OSD attributes lists

7.1.3.1 Attributes lists overview

…

The LIST TYPE field (see table 127) specifies the format of all attributes list entries in the attributes list.

Table 127 — List type values

List Type	Description	Support Requirement	Reference	Allowed Use		
				Get Attributes		Set Attributes List
				List	Response	
0h	Reserved			No	No	No
1h	Retrieve attributes for ~~this~~ the specified OSD object	Mandatory	7.1.3.2	Yes	No	No
2h - 8h	Reserved			No	No	No
9h	Retrieved/Set attributes for ~~this~~ the specified OSD object	Mandatory	7.1.3.3	No	Yes	Yes
Ah - Eh	Reserved			No	No	No
Fh	Retrieved attributes for a CREATE command (see 6.3) that creates more than one user object	Mandatory	7.1.3.4	No	Yes	No

If list type 1h (see 7.1.3.2) is used to retrieve attributes for ~~this~~ the specified OSD object, the list type of the list containing the retrieved objects shall be:

a) Fh (see 7.1.3.4) for a CREATE command that creates more than one user object; or
b) 9h (see 7.1.3.3) for all other commands and for a CREATE command that creates only one user object.

The LIST LENGTH field indicates the number of bytes of attributes list entries that follow. The LIST LENGTH field may contain zero.

For an attributes list sent from the device server to the application client, a list length of zero indicates that all of the requested attributes have an attribute length of zero.

The application client should set the list length to zero in any attributes list that it sends to the device server. The device server shall use the length of the list specified in the CDB and shall ignore the contents of the LIST LENGTH field.

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

The attributes list entry format shown in table 128 is used for specifying the attributes to be retrieved by a GET ATTRIBUTES command (see 6.12) or equivalent command function.

Table 128 — List entry format for retrieving attributes for ~~this~~ the specified OSD object

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			ATTRIBUTES PAGE				
3								(LSB)
4	(MSB)			ATTRIBUTE NUMBER				
7								(LSB)

The ATTRIBUTES PAGE field specifies the page number of one attribute to be returned. If the specified attributes page number is not associated with ~~the user~~ an object specified by a command, the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

NOTE n1 Some commands (e.g., LIST (see 6.13)) define methods that allow the attributes in multiple objects to be processed by a single command. The conditions under which an attributes page number is not required to be associated with the object specified by a command's CDB appear in the command definition subclauses for the exceptional commands.

The ATTRIBUTE NUMBER field specifies:

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

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

Specifying attributes page and attribute ~~numbers~~ number values of FFFF FFFFh causes all defined attributes values in all defined pages associated with the OSD object specified by a command to be returned. Specifying an attribute numbers value of FFFF FFFFh causes all defined attributes values in the specified attributes page to be returned.

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

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

The attributes list entry format shown in table 129 is used for returning the each attribute value to be retrieved by a GET ATTRIBUTES command (see 6.12) and for specifying each attribute value to be set by a SET ATTRIBUTES command (see 6.21) or equivalent command functions.

Table 129 — List entry format for retrieved attributes and for setting attributes for ~~this~~ the specified OSD object

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)			ATTRIBUTES PAGE				
3								(LSB)
4	(MSB)			ATTRIBUTE NUMBER				
7								(LSB)
8	(MSB)			ATTRIBUTE LENGTH (n-9)				
9								(LSB)
10	(MSB)			ATTRIBUTE VALUE				
n								(LSB)

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

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

The ATTRIBUTE LENGTH field specifies the length of the attribute value in bytes.

The ATTRIBUTE VALUE field specifies the attribute value.

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

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

… No other changes in 7.1.3