ENDL
TEXAS

Date: 18 September 2005
To: T10 Technical Committee & SNIA OSD TWG
From: Ralph O. Weber
Subject: OSD-2 Four New Multi-Object Commands

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.

This proposal defines four new commands that operate on multiple objects. Document T10/05-316 describes multi-object enhancements to the LIST and LIST COLLECTION commands.

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 Make minor editorial changes requested by the T10 CAP working group

The document of reference for all clause, subclause, table, and page numbers is osd2r00.pdf.

Additions Proposed in OSD-2 (changes in this section are not highlighted because all text is new)

6.w GET MEMBER ATTRIBUTES

The GET MEMBER ATTRIBUTES command (see table x1) 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 4.7.2). The GET MEMBER ATTRIBUTES command is a multi-object command (see 4.6.6.2).

Table x1 — GET MEMBER ATTRIBUTES command

Bit Byte	7	6	5	4	3	2	1	0
8	(MSB)			SERVICE ACTION (8822h)				
9								(LSB)
10				OPTIONS BYTE				
11	Reserved		GET/SET CDBFMT		Reserved			
12				TIMESTAMPS CONTROL				
13				Reserved				
15								
16	(MSB)			PARTITION_ID				
23								(LSB)
24	(MSB)			COLLECTION_OBJECT_ID				
31								(LSB)
32				Reserved				
51								
52				Get and set attributes parameters (see 5.2.2)				
79								
80				Capability (see 4.9.2.2)				
159								
160				Security parameters (see 5.2.6)				
199								

The contents of the OPTIONS BYTE field are defined in 5.2.4.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2. Page format attribute processing is illegal for the GET MEMBER ATTRIBUTES command. If 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. The device server shall constrain the Collection_Object_ID values as defined in 4.6.6.2.

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. Get and set attributes processing requirements specific to multi-object commands are defined in 4.6.6.2.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

6.x QUERY

6.x.1 Introduction

The QUERY command (see table x2) 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 x2 — QUERY command

Bit Byte	7	6	5	4	3	2	1	0
8	(MSB)	SERVICE ACTION (8820h)						
9								(LSB)
10	Reserved							
11	Reserved		GET/SET CDBFMT		Reserved			
12	TIMESTAMPS CONTROL							
13	Reserved							
15								
16	(MSB)	PARTITION_ID						
23								(LSB)
24	(MSB)	COLLECTION_OBJECT_ID						
31								(LSB)
32	(MSB)	QUERY LIST LENGTH						
35								(LSB)
36	(MSB)	ALLOCATION LENGTH						
43								(LSB)
44	Reserved							
51								
52	Get and set attributes parameters (see 5.2.2)							
79								
80	Capability (see 4.9.2.2)							
159								
160	Security parameters (see 5.2.6)							
199								

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

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

The contents of the PARTITION_ID field are defined in 5.2.5.

The COLLECTION_OBJECT_ID field specifies Collection_Object_ID (see 4.6.6) to be processed. The device server shall constrain the Collection_Object_ID values as defined in 4.6.6.2.

The QUERY LIST LENGTH field specifies the number of bytes to be transferred of query list data (see 6.x.2) that contain the attributes query criteria

The ALLOCATION LENGTH field specifies the maximum number of bytes that an application client has allocated for the matches list (see 6.x.3). 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 matches list data that is 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 matches list is truncated, the contents of the ADDITIONAL LENGTH field (see 6.x.3) shall not be altered to reflect the truncation.

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. Get and set attributes processing requirements specific to multi-object commands are defined in 4.6.6.2.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

6.x.2 Query list format

The query list (see table x5) specifies the criteria for selecting the user objects whose User_Object_IDs are returned in the matches list.

Table x3 — Query list format

Bit **Byte**	7	6	5	4	3	2	1	0
0	Reserved				QUERY TYPE			
1	Reserved							
3								
	Query criteria entries							
4	Query criteria entry 0 (see table x5)							
	⋮							
	Query criteria entry x (see table x5)							
n								

The QUERY TYPE field (see table x4) specifies the format of the query criteria entries that follow.

Table x4 — Query type codes

Value	Description
0h	A match with any query criteria entry shall cause the user object to appear in the list.
1h	Matching all query criteria entries shall cause the user object to appear in the list.
2h to Fh	Reserved

Each query criteria entry (see table x5) specifies matching criteria for one attribute.

Table x5 — Query criteria entry format

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved							
1								
2	(MSB)	QUERY ENTRY LENGTH (n-3)						
3								(LSB)
4	(MSB)	ATTRIBUTES PAGE						
7								(LSB)
8	(MSB)	ATTRIBUTE NUMBER						
11								(LSB)
12	(MSB)	MINIMUM ATTRIBUTE VALUE LENGTH (m-13)						
13								(LSB)
14	(MSB)	MINIMUM ATTRIBUTE VALUE						
m								(LSB)
m+1	(MSB)	MAXIMUM ATTRIBUTE VALUE LENGTH (n-m-2)						
m+2								(LSB)
m+3	(MSB)	MAXIMUM ATTRIBUTE VALUE						
n								(LSB)

The QUERY ENTRY LENGTH field specifies the number of bytes that follow in the query entry.

The ATTRIBUTES PAGE field specifies the page number of the attribute value. If the attributes page is not between 0h and 2FFF FFFFh, inclusive, the command shall be terminated with a CHECK CONDITION, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

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

The MINIMUM ATTRIBUTE VALUE LENGTH field specifies the number of bytes that follow in the MINIMUM ATTRIBUTE VALUE field.

The MINIMUM ATTRIBUTE VALUE field specifies the minimum attribute value necessary for a user object to meet the criteria.

The MAXIMUM ATTRIBUTE VALUE LENGTH field specifies the number of bytes that follow in the MAXIMUM ATTRIBUTE VALUE field.

The MAXIMUM ATTRIBUTE VALUE field specifies the maximum attribute value necessary for a user object to meet the criteria.

6.x.3 Matches list format

The matches list (see table x6) contains the User_Object_ID for every user object that matched the query criteria.

Table x6 — Matches list format

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	ADDITIONAL LENGTH (n-7)						
7								(LSB)
8	Reserved							
11								
12	OBJECT DESCRIPTOR FORMAT (21h)						Reserved	
	Object descriptor list							
13	Object descriptor (first)							
	⋮							
	Object descriptor (last)							
n								

The ADDITIONAL LENGTH field indicates the number of bytes of matches list data that follow. If the matches list is truncated due to insufficient allocation length (see 6.x.1), 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 OBJECT DESCRIPTOR FORMAT field shall contain 21h indicating that the object descriptors have the format shown in 6.13.3.5.

Each object descriptor (see 6.13.3.5) contains the User_Object_ID of one user object that matches the query criteria in the query list (see 6.x.2).

Note: 6.13.3.5 is defined by proposal T10/05-316.

6.y REMOVE MEMBER OBJECTS

The REMOVE MEMBER OBJECTS command (see table x7) 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 x7 — REMOVE MEMBER OBJECTS command

Bit Byte	7	6	5	4	3	2	1	0
8	(MSB)	SERVICE ACTION (8821h)						
9								(LSB)
10	OPTIONS BYTE							
11	Reserved		GET/SET CDBFMT		Reserved			
12	TIMESTAMPS CONTROL							
13	Reserved							
15								
16	(MSB)	PARTITION_ID						
23								(LSB)
24	(MSB)	COLLECTION_OBJECT_ID						
31								(LSB)
32	Reserved							
51								
52	Get and set attributes parameters (see 5.2.2)							
79								
80	Capability (see 4.9.2.2)							
159								
160	Security parameters (see 5.2.6)							
199								

The contents of the OPTIONS BYTE field are defined in 5.2.4.

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

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

The contents of the PARTITION_ID field are defined in 5.2.5.

The COLLECTION_OBJECT_ID field specifies Collection_Object_ID (see 4.6.6) to be processed. The device server shall constrain the Collection_Object_ID values as defined in 4.6.6.2.

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. Get and set attributes processing requirements specific to multi-object commands are defined in 4.6.6.2.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

6.z SET MEMBER ATTRIBUTES

The SET MEMBER ATTRIBUTES command (see table x8) 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 4.7.2). The SET MEMBER ATTRIBUTES command is a multi-object command (see 4.6.6.2).

Table x8 — SET MEMBER ATTRIBUTES command

Bit Byte	7	6	5	4	3	2	1	0
8	(MSB)	SERVICE ACTION (8823h)						
9								(LSB)
10	OPTIONS BYTE							
11	Reserved		GET/SET CDBFMT		Reserved			
12	TIMESTAMPS CONTROL							
13	Reserved							
15								
16	(MSB)	PARTITION_ID						
23								(LSB)
24	(MSB)	COLLECTION_OBJECT_ID						
31								(LSB)
32	Reserved							
51								
52	Get and set attributes parameters (see 5.2.2)							
79								
80	Capability (see 4.9.2.2)							
159								
160	Security parameters (see 5.2.6)							
199								

The contents of the OPTIONS BYTE field are defined in 5.2.4.

The GET/SET CDBFMT field specifies the format of the get and set attributes parameters as described in 5.2.2. Page format attribute processing is illegal for the SET MEMBER ATTRIBUTES command. If 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. The device server shall constrain the Collection_Object_ID values as defined in 4.6.6.2.

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. Get and set attributes processing requirements specific to multi-object commands are defined in 4.6.6.2.

The capability is defined in 4.9.2.2.

The security parameters are defined in 5.2.6.

Changes Proposed in OSD-2

Additions and some table changes are shown in red, removals in ~~blue strikeout~~, and editing instructions in green.

Modify 4.6.6 as follows.

4.6.6 Collections

4.6.6.1 Overview

Support for collections is optional. If collections are not supported:

a) The length of attribute number 4h in the User Object Directory attributes page (see 7.1.2.7) shall be zero for every user object (i.e., no Collections attributes pages identified); and
b) Zero shall be returned as the length of attribute number 0h in every Collections attributes page (see 7.1.2.19).

A partition may contain zero or more collections each of which may contain zero or more user objects. One user object may be a member of zero or more collections. If the collection type attribute in the Collection Information attributes page (see 7.1.2.10) contains 00h, user ~~User~~ objects are added to or removed from the membership of a collection by setting attribute values in the user object's Collections attributes page (see 7.1.2.19).

Collections have the Partition_ID of the partition to which they belong and a Collection_Object_ID (see 4.6.2) that is assigned by the OSD logical unit when the collection is created. A collection is a member of only one partition.

Within a single partition, no collection shall be assigned the same Collection_Object_ID as any User_Object_ID and no user object shall be assigned the same User_Object_ID value as any Collection_Object_ID (i.e., collections and user objects share the same number space for their identifier values).

A collection is created using the CREATE COLLECTION command (see 6.5) and deleted using the REMOVE COLLECTION command (see 6.19). The page format of the Collections attributes page (see 7.1.2.19) lists all the collections in which a user object is a member. The LIST COLLECTION command (see 6.14) lists all the collections in a partition or all the user objects that are members of a collection.

A collection does not contain a read/write data area. The device server shall terminate all READ commands, WRITE commands, and APPEND commands sent to the collection with a CHECK CONDITION status, setting the sense key to ILLEGAL REQUEST and the additional sense code to INVALID FIELD IN CDB.

4.6.6.2 Commands that use collections to affect multiple user objects

Commands such as SET MEMBER ATTRIBUTES (see 6.z) (i.e., multi-object commands) process multiple user objects using the membership of a collection as a dynamic list of the user objects on which the specified operations are to be performed.

Multi-object commands process only collections whose collection type attribute contains 01h. If the COLLECTION_OBJECT_ID field in a multi-object command CDB specifies a collection for which the collection type attribute in the Collection Information attributes page (see 7.1.2.10) contains a value other than 01h, the command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.

If the COLLECTION_OBJECT_ID field in a multi-object command CDB specifies a object that is not a collection, 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.

After the specified operations have been successfully completed on a user object, that user object shall be removed from the specified collection. As a result of this requirement, the following conditions apply:

a) After an error condition that prevented processing of all user objects in the collection is corrected, the same command specifying the same collection may be sent to continue processing;
b) Application clients may poll to determine the progress of a multi-object command using the LIST COLLECTION command (see 6.14); and
c) Application clients may poll to determine the progress of a multi-object command by retrieving the number of members attribute value in the Collection Information attributes page (see 7.1.2.10).

NOTE n1 The LIST COLLECTION command is not a multi-object command.

Two multi-object commands shall not concurrently process the same collection. If a multi-object command is received with the COLLECTION_OBJECT_ID field in the CDB specifying the Collection_Object_ID (see 4.6.2) of a collection that is already being processed by a different multi-object command, 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 device sever may process more than one user object concurrently.

If an error is detected during the processing of a user object:

a) The user object shall not be removed from the collection;
b) Processing that has already been started on any other user object shall be completed to the greatest degree possible and any user objects for which processing is successfully completed shall be removed from the collection;
c) The policy access tag attribute in the User Object Policy/Security attributes page (see 7.1.2.23) for any user object for which an error is detected shall be updated as described in 4.9.3;
d) Processing shall not be started for any user object that has not already started processing; and
e) When no user objects are being processed, the command shall be terminated with the status and sense data corresponding to the first error that was detected.

If a multi-object command is terminated as part of processing any task management function, the device server shall either:

a) Establish a consistent, stable state for each user object being processed; or
b) Set the policy access tag attribute in the User Object Policy/Security attributes page described in 4.9.3 for any user object for which it is not possible to establish consistent state.

The device server shall not remove the specified collection upon completion of the multi-object command, even if the collection contains zero user objects.

If the CDB GET/SET CDBFMT field contains 11b (i.e., when list format attributes processing is specified), multi-object commands allow setting and retrieving of both collection attributes and user object attributes. The get and set attributes parameters are defined in 5.2.2, the list format is defined in 7.1.3, attribute page number values in the get attributes list shall be processed as shown in table 2a, and attribute page numbers in the set attributes list shall be processed as shown in table 2b.

Table 2a — Attributes retrieval requirements for multi-object commands

Attribute page number values	Command	Description
C+0h to C+2FFF FFFFh and F000 0000h to FFFF FFFFh	Any multi-object command	The attribute values shall be returned in the retrieved attributes segment of the Data-In Buffer (see 4.12.3) as defined in 5.2.2.3 using list type Fh (see 7.1.3.4).
0h to 2FFF FFFFh		The attribute values for every user object that is a member of the collection shall be returned in the retrieved attributes segment of the Data-In Buffer as defined in 5.2.2.3 using list type Fh (see 7.1.3.4).
P+0h to P+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.

Table 2b — Attributes setting requirements for multi-object commands

Attribute page number values	Command	Description
C+0h to C+2FFF FFFFh and F000 0000h to FFFF FFFFh	Any multi-object command	The setting of attributes shall be processed as defined in 5.2.2.3.
0h to 2FFF FFFFh	GET MEMBER ATTRIBUTES or SET MEMBER ATTRIBUTES	The setting of attributes shall be processed as defined in 5.2.2.3 and the same user object attribute values shall be set in every user object that is a member of the collection.
	QUERY or REMOVE MEMBER OBJECTS	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.
P+0h to P+2FFF FFFFh and R+0h to R+2FFF FFFFh	Any multi-object command	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.

Multi-object commands may allow retrieval and setting of attribute values using page format (see 7.1.2), but only collection attribute pages shall be processed in page format.

Modify 4.7.2 as follows:

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

OSD commands provide the application client with the ability to get and set attributes as part of processing the command (e.g., a WRITE command may also retrieve the user object logical length attribute). This subclause defines the relative order of the command functions (see 3.1.10) processing within a single command.

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

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

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

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

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

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

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

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

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

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

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

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

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							

Note: the M_OBJECT bit and QUERY bit are proposed by document T10/05-316 but applies to several commands in this proposal. The proposed definitions in this proposal are newer than those in T10/05-316.

…

A multiple objects (M_OBJECT) bit set to one in combination with other permissions bits allows the member user objects in a collection to be processed as a group (e.g., retrieving attributes from multiple user objects, setting attributes in multiple user objects, and removing multiple user objects). An M_OBJECT bit set to zero prohibits the processing of member user object in a collection.

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.

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 GET MEMBER ATTRIBUTES command addressed to a collection	COLLECTION	see table 16	U/C
⋮	⋮	⋮	⋮
A QUERY command addressed to a collection	COLLECTION	QUERY	U/C
⋮	⋮	⋮	⋮
A REMOVE MEMBER OBJECTS command addressed to a collection	COLLECTION	REMOVE and M_OBJECT	U/C
⋮	⋮	⋮	⋮
A SET MEMBER ATTRIBUTES command addressed to a collection	COLLECTION	see table 16	U/C
⋮	⋮	⋮	⋮
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

	Capability Field values that allow attribute-related functions		
Attribute-Related Functions Allowed	**Object Type Name**	**Permission Bits That Are Set To One**	**Object Descriptor Name**
⋮	⋮	⋮	⋮
As part of a GET MEMBER ATTRIBUTES command, QUERY command, REMOVE MEMBER OBJECTS command, or SET MEMBER ATTRIBUTES, the retrieval of attributes from each user object in a collection	COLLECTION	GET_ATTR and M_OBJECT	U/C
⋮	⋮	⋮	⋮
As part of a GET MEMBER ATTRIBUTES command or SET MEMBER ATTRIBUTES, the setting of attributes from each user object in the collection, except attributes in a User Object Policy/ Security attributes page (see 7.1.2.23)	COLLECTION	SET_ATTR and M_OBJECT	U/C
As part of a GET MEMBER ATTRIBUTES command SET MEMBER ATTRIBUTES, the setting of attributes from any user object in the collection	COLLECTION	SET_ATTR, M_OBJECT, and POL/SEC	U/C
⋮	⋮	⋮	⋮
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 Table 39 as follows.

Table 39 — OSD commands that are allowed in the presence of various reservations

OSD Command	Addressed LU 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 I_T nexus not registered	
	Write Excl	Excl Access		Write Excl RR	Excl Access – RR
⋮	⋮	⋮	⋮	⋮	⋮
GET ATTRIBUTES	Allowed	Conflict	Allowed	Allowed	Conflict
GET MEMBER ATTRIBUTES	Allowed	Conflict	Allowed	Allowed	Conflict
LIST	Allowed	Conflict	Allowed	Allowed	Conflict
LIST COLLECTION	Allowed	Conflict	Allowed	Allowed	Conflict
⋮	⋮	⋮	⋮	⋮	⋮
QUERY	Allowed	Conflict	Allowed	Allowed	Conflict
READ	Allowed	Conflict	Allowed	Allowed	Conflict
REMOVE	Conflict	Conflict	Allowed	Conflict	Conflict
REMOVE COLLECTION	Conflict	Conflict	Allowed	Conflict	Conflict
REMOVE MEMBER OBJECTS	Conflict	Conflict	Allowed	Conflict	Conflict
REMOVE PARTITION	Conflict	Conflict	Allowed	Conflict	Conflict
SET ATTRIBUTES	Conflict	Conflict	Allowed	Conflict	Conflict
SET MEMBER ATTRIBUTES	Conflict	Conflict	Allowed	Conflict	Conflict
⋮	⋮	⋮	⋮	⋮	⋮
Any command that retrieves attributes	see the command entry in this table				
Any command that sets attributes	Conflict	Conflict	Allowed	Conflict	Conflict
Key: **LU**=Logical Unit, **Excl**=Exclusive, **RR**=Registrants Only or All Registrants					

Modify Table 48 as follows.

Table 40 — Commands for OSD type devices

Command name	Operation code	Service action [a]	Type	Reference
⋮	⋮	⋮	⋮	⋮
CREATE COLLECTION	7Fh	8815h	O [b]	6.5
⋮	⋮	⋮	⋮	⋮
FLUSH COLLECTION	7Fh	881Ah	~~M~~O [b]	6.8
⋮	⋮	⋮	⋮	⋮
GET MEMBER ATTRIBUTES	7Fh	8822h	O [b]	6.w
⋮	⋮	⋮	⋮	⋮
LIST COLLECTION	7Fh	8817h	O	6.14
⋮	⋮	⋮	⋮	⋮
QUERY	7Fh	8820h	O [b]	6.x
⋮	⋮	⋮	⋮	⋮
REMOVE COLLECTION	7Fh	8816h	O [b]	6.19
REMOVE MEMBER OBJECTS	7Fh	8821h	O [b]	6.y
⋮	⋮	⋮	⋮	⋮
SET MEMBER ATTRIBUTES	7Fh	8823h	O [b]	6.z
⋮	⋮	⋮	⋮	⋮

Type Key: M = Command implementation is mandatory.
O = Command implementation is optional.
X = Command implementation requirements given in SPC-3.

[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] Support for this command is mandatory if collections are supported (see 4.6.6).

[c] Unless the security method in effect for the root object and every partition in the OSD logical unit is NOSEC (see 4.10.1), this 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 INVALID COMMAND OPERATION CODE. If the security method in effect for the root object or any partition in the OSD logical unit is not NOSEC, this command may be performed only by using the PERFORM SCSI COMMAND command (see 6.15).

[d] The effects on established persistent reservations and registrations if the security method in effect for the root object or any partition changes from NOSEC to any other security method are described in 4.16.

Modify 7.1.2.10 as follows.

7.1.2.10 Collection Information attributes page

The Collection Information attributes page (C+1h) shall contain the attributes listed in table 93.

Table 93 — 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
~~Ah~~ Ch to 80h		Reserved	No	
81h	8	Used capacity	No	Yes
82h 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 Information".

The Partition_ID attribute (number 1h) shall contain the Partition_ID of the collection with which the Collection Information attributes page is associated.

The Collection_Object_ID attribute (number 2h) shall contain the Collection_Object_ID (see 4.6.6) of the collection with which the Collection Information attributes page is associated.

The username attribute (number 9h) shall contain a binary valued identification of the user for the collection specified by the application client. A CREATE COLLECTION command (see 6.5) shall copy the username attribute from the Partition Information attributes page (see 7.1.2.9) to the new Collection Information attributes page.

The collection type attribute (number Ah) shall identify the characteristics (see table 93a) of the collection.

Table 93a — Collection type codes

Value	Description
00h	User objects may be added to or removed from the collection using the Collections attributes page (see 7.1.2.19).
01h	User objects shall not be added or removed from the collection except as part of processing multi-object commands (see 4.6.6.2).
02h to FFh	Reserved

The number of members attribute (number Bh) shall indicate the number of user objects that are members of the collection.

The used capacity attribute (number 81h) shall contain the number of bytes used by the collection including attributes bytes.

If a set attributes list (see 5.2.2.3) contains an entry specifying the number of an attribute that table 93 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 5.2.2.2) specifies the number of an attribute that table 93 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.

Modify 7.1.2.19 as follows.

7.1.2.19 Collections attributes page

The Collections attributes page (4h) shall contain the attributes listed in table 110.

Table 110 — Collections attributes page contents

Attribute Number	**Length** (bytes)	**Attribute**	**Application Client Settable**	**OSD Logical Unit Provided**
0h	0 or 40	Page identification	No	Yes
1h to FFFF FF00h	0 or 8	Collection pointer	Yes/No [a]	No
FFFF FF01h to FFFF FFFEh		Reserved	No	

[a] If the collection type attribute in the Collection Information attributes page (see 7.1.2.10) contains 00h, the Collection pointer attribute shall be application client settable. If the collection type attribute in the Collection Information attributes page contains 01h, the Collection pointer attribute shall not be application client settable.

If collections are supported, 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 Collections". If collections are not supported, the length of the page identification attribute shall be zero.

Each collection pointer attribute (1h to FFFF FF00h) may be:

a) A zero length attribute (i.e., contain no value); or
b) The Collection_Object_ID of a collection (see 4.6.6) to which the user object belongs.

If the collection type attribute in the Collection Information attributes page contains 00h, a ~~A~~ user object is made a member of a collection by setting one of its collection pointer attribute values to the Collection_Object_ID of that collection.

If the collection type attribute in the Collection Information attributes page contains 00h, a ~~A~~ user object is removed from the membership of a collection by:

a) Changing the collection pointer attribute identifying that collection to have a length of zero; or
b) Setting the collection pointer attribute identifying that collection to the Collection_Object_ID of a different collection.

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 a set attributes list (see 5.2.2.3) contains an entry that sets:

a) The same Collection_Object_ID in more than one collection pointer attribute;
b) A collection pointer attribute to a value that is not a Collection_Object_ID; or
c) A collection pointer attribute to any length other than zero or eight.

The command shall be terminated with a CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB if:

a) The CDB SET ATTRIBUTE NUMBER field and the set attributes data specified by the SET ATTRIBUTES OFFSET field (see 5.2.2.2) sets:
   A) The same Collection_Object_ID in more than one collection pointer attribute; or
   B) A collection pointer attribute to a value that is not a Collection_Object_ID;
   or
b) The CDB SET ATTRIBUTE LENGTH field contains a value other than zero or eight.

If setting a collection pointer attribute causes the number of collection pointer attributes with non-zero attribute lengths to exceed the value in the collections per user object attribute in the Partition Quotas attributes page (see 7.1.2.13), then a quota error shall be generated (see 4.8.2). The quota testing principles described in 4.8.3 apply to the testing of the object count quota.

If a set attributes list contains an entry specifying the number of an attribute that table 110 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 specifies the number of an attribute that table 110 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.

Modify Table B.1 as follows.

Table B.1 — Numerical order OSD service action codes

Service Action	Command
⋮	⋮
881Dh to 881Fh	Reserved
8820h	QUERY
8821h	REMOVE MEMBER OBJECTS
8822h	GET MEMBER ATTRIBUTES
8823h	SET MEMBER ATTRIBUTES
8824h to 8F7Dh	Reserved
⋮	⋮

Modify Table C.1 as follows.

Table C.1 — Numerical order attributes defined by this standard

Page Number	Page Name	Attribute Number	Attribute
⋮	⋮	⋮	⋮
6000 0001h	Collection Information	0h	Page identification
		1h	Partition_ID
		2h	Collection_Object_ID
		9h	Username
		Ah	Collection type
		Bh	Number of members
		81h	Used capacity
⋮	⋮	⋮	⋮