ENDL
TEXAS

Date: 26 June 2004
To: T10 Technical Committee and SNIA OSD TWG
From: Ralph O. Weber
Subject: OSD Error Reporting and Sense Data Descriptors

This proposal describes changes to be made in OSD r09 to:

a) Move OSD-specific sense data descriptor format definitions out of SPC-3 and in to the OSD draft;
b) Define the sense data descriptor used to report which quota has been exceeded;
c) Rename the OSD object identification sense data descriptor to something more generic; and
d) Define how the renamed OSD object identification sense data descriptor is used to indicate which command functions had been processed and which not at the time the error was detected.

The changes in this proposal are intended to address the following OSD Letter Ballot comments (see T10/04-108): IBM 36), and IBM 63).

Revision History

r0 Original revision
r1 Remove specification that credential validation and capability validation are processed first because this needs to be specified in the security model. Add detailed discussion of how the OSD error identification sense data descriptor indicates which command functions had been processed and which not at the time the error was detected.
r2 Remove comment Panasas 2) from the list of comments addressed by this proposal, since that is no longer true. Swap the column positions in the "Command functions bits combinations" table to clarify that the emphasis is not on the bit values.
r3 Clarify NOT INITIATED COMMAND FUNCTIONS field and COMPLETED COMMAND FUNCTIONS field definitions. Based on the clarification, swap the column positions in the "Command functions bits combinations" table back to their r1 positions.
r4 Reflect the SNIA OSD TWG agreement to allow capabilities to be validated immediately prior to each command function processing step by adding capability validation bits for each step. Since the number of used bits has doubled, double the size of the bit mask fields too.

Differences from r0 are indicated by change bars.

Detailed OSD r09 Changes

For completeness, accepted changes from the following OSD Letter Ballot comments (see T10/04-108) are included in this document: AMCC 4), IBM 35), and IBM 64).

Text already appearing in OSD r09 is shown in black. Text appearing in OSD r09 that is to be removed is shown in red strike through. Text to be added is shown in blue.

4.8.2 Quota errors

If one of the quota error conditions described in 5.2.1 and clause 6 occurs, ~~processing of~~ the command shall be terminated with a CHECK CONDITION status, with the sense key set to DATA PROTECT and the additional sense code set to QUOTA ERROR. ~~and a quota error shall be reported as follows:~~

~~a) The status shall be CHECK CONDITION;~~
~~b) The sense key shall be DATA PROTECT;~~
~~c) The additional sense code shall be QUOTA ERROR; and~~
~~d) The sense data shall include the OSD attribute identification sense data descriptor (see SPC-3) with one or more attribute descriptors identifying the quota attribute or attributes that have been exceeded.~~

The sense data shall include the OSD attribute identification sense data descriptor (see 4.13.2.3) with one or more attribute descriptors identifying the quota attribute or attributes that have been exceeded.

...

4.9.3.4 The CMDRSP security method

...

If the credential and capabilities validation process successfully validates the integrity check value associated with the command, the device server shall:

1) Compute an integrity check value for the response data using:
   A) The algorithm specified in the capability INTEGRITY CHECK VALUE ALGORITHM field (see 4.9.4.3);
   B) The following array of bytes:
      1) The request nonce from the CDB (see 5.2.5);
      2) The status byte; and
      3) If the status is CHECK CONDITION, the sense data with the RESPONSE INTEGRITY CHECK VALUE field in the OSD response integrity check value sense data descriptor (see 4.13.2.2) ~~(see SPC-3)~~ set to zero;

      and
   C) The capability key (see 4.9.4.2) for the reconstructed credential (see 4.9.5.3);

   and
2) Place the computed integrity check value in the following location:
   A) If the status is not CHECK CONDITION, the computed integrity check value shall be placed in the response integrity check value attribute in the Current Command attributes page (see 7.1.2.24); or
   B) If the status is CHECK CONDITION, the computed integrity check value shall be placed in the RESPONSE INTEGRITY CHECK VALUE field in the OSD response integrity check value sense data descriptor (see 4.13.2.2) in the sense data.

...

4.13 Error reporting

4.13.1 Introduction

OSD logical units shall use descriptor format sense data (see SPC-3) to report all errors.

All sense data returned by OSD device servers shall include the OSD ~~object~~ error identification sense data descriptor (see 4.13.2.1) ~~(see SPC-3)~~ to identify the OSD object in which the reported error was detected.

If it is possible to identify a specific byte or range of bytes within a user object as being associated with an error, the information sense data descriptor (see SPC-3) shall be included in the sense data with the INFORMATION field set to the byte within the user object associated with the error or the first byte in the range of bytes within the user object associated with the error.

If the CMDRSP security method or the ALLDATA security method (see 4.9.3) is used to process the command, the sense data shall include the OSD response integrity check value sense data descriptor (see 4.13.2.2). ~~(see SPC-3) with the RESPONSE INTEGRITY CHECK VALUE field containing an integrity check value (see 4.9.7) that is computed as described in 4.9.3.4.~~

> ~~NOTE 5 If the status is not CHECK CONDITION and no sense data is transferred, the response integrity check value is returned in the response integrity check value attribute in the Current Command attributes page (see 7.1.2.24).~~

If the status is not CHECK CONDITION and no sense data is transferred, the response integrity check value is returned in the response integrity check value attribute in the Current Command attributes page (see 7.1.2.24).

The OSD CDB is very large. To reduce uncertainty in determining errors in CDB field settings or in parameter data, any sense data having the sense key set to ILLEGAL REQUEST should include the sense key specific sense data descriptor (see SPC-3) with the field pointer sense key specific data.

Errors other than those defined in this standard may be reported as needed. The sense data shall include the appropriate sense key and additional sense code (see SPC-3) to identify the condition.

Errors may occur after the command has completed. For such errors, SPC-3 defines a deferred error reporting mechanism.

4.13.2 OSD-specific sense data descriptors

4.13.2.1 OSD error identification sense data descriptor

The OSD error identification sense data descriptor (see table x1) provides information that identifies the OSD object associated with the error reported in the sense data.

Table x1 — OSD error identification sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (06h)							
1	ADDITIONAL LENGTH (1Eh)							
2	Reserved							
7								
8	NOT INITIATED COMMAND FUNCTIONS							
11								
12	COMPLETED COMMAND FUNCTIONS							
15								
16	(MSB)	PARTITION_ID						
23								(LSB)
24	(MSB)	USER_OBJECT_ID						
31								(LSB)

The NOT INITIATED COMMAND FUNCTIONS field contains the command functions bits (see table x2) that indicate (see table x4) which command functions had not been initiated at the time the error reported in the sense data was detected.

The COMPLETED COMMAND FUNCTIONS field contains the command functions bits (see table x2) that indicate (see table x4) which command functions had been completed at the time the error reported in the sense data was detected.

The PARTITION_ID field contains the Partition_ID (see 4.6.4) of the partition that is associated with the error being reported.

The OBJECT_ID field contains the Collection_Object_ID (see 4.6.6) or User_Object_ID (see 4.6.5) of the object that is associated with the error being reported.

The command functions bits (see table x2) are contained in the NOT INITIATED COMMAND FUNCTIONS field and COMPLETED COMMAND FUNCTIONS field (see table x1).

Table x2 — Command functions bits

Bit Byte	7	6	5	4	3	2	1	0
0	VALIDATION	Reserved	CMD_CAP_V	COMMAND	Reserved	Reserved	Reserved	Reserved
1	Reserved	Reserved	Reserved	IMP_ST_ATT	Reserved	Reserved	Reserved	Reserved
2	Reserved	Reserved	SA_CAP_V	SET_ATT	Reserved	Reserved	Reserved	Reserved
3	Reserved	Reserved	GA_CAP_V	GET_ATT	Reserved	Reserved	Reserved	Reserved

The command functions bits and the command functions that they indicate are listed in table x3.

Table x3 — Command functions indicated by the command functions bits

Command functions bit	Command function indicated
VALIDATION	Validation of the command, including security parameters
CMD_CAP_V	Capability verification for those command functions not related to attributes (e.g., writing data to a user object)
COMMAND	Processing of those command functions not related to attributes
IMP_ST_ATT	Processing of any set attributes command functions resulting from the processing of the command (e.g.,changes due to a WRITE command)
SA_CAP_V	Capability verification for all set attributes command functions specified in the CDB
SET_ATT	Processing of any set attributes command functions specified in the CDB
GA_CAP_V	Capability verification for all get attributes command functions specified in the CDB
GET_ATT	Processing of any get attributes command functions specified in the CDB

The interpretation of the combinations of the command functions bits in the NOT INITIATED COMMAND FUNCTIONS field and COMPLETED COMMAND FUNCTIONS field is shown in table x4.

Table x4 — Command functions bits combinations

NOT INITIATED COMMAND FUNCTIONS field	COMPLETED COMMAND FUNCTIONS field	Status of the indicated command function at the time the error reported by the sense data was detected
1	0	Processing was requested, but was not initiated and not completed
0	0	Processing was not requested, or processing was in progress
0	1	Processing was requested and completed
1	1	Reserved

4.13.2.2 OSD response integrity check value sense data descriptor

The OSD response integrity check value sense data descriptor (see table x5) contains the response integrity check value used when the OSD security method is CMDRSP or ALLDATA.

Table x5 — OSD response integrity check value sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (07h)							
1	ADDITIONAL LENGTH (0Ch)							
2	(MSB)	RESPONSE INTEGRITY CHECK VALUE						
13								(LSB)

The RESPONSE INTEGRITY CHECK VALUE field contains the response integrity check value (see 4.9.7) that is computed as described in 4.9.3.4 for the command for which the error being reported.

4.13.2.3 OSD attribute identification sense data descriptor

The OSD attribute identification sense data descriptor (see table x6) identifies one or more attributes (see 7.1) associated with the error reported in the sense data.

Table x6 — OSD attribute identification sense data descriptor format

Bit Byte	7	6	5	4	3	2	1	0
0	DESCRIPTOR TYPE (08h)							
1	ADDITIONAL LENGTH (n-2)							
2	Reserved							
3	Reserved							
	Attribute descriptors							
4	Attribute descriptor 0 (see table x7)							
	⋮							
	Attribute descriptor x (see table x7)							
n								

Each attribute descriptor (see table x7) identifies one attribute associated with the error reported in the sense data.

Table x7 — Sense data attribute descriptor format

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

The ATTRIBUTE PAGE field contains the attribute page number (see 4.7.3) for the attributes page containing the attribute associated with the error reported in the sense data.

The ATTRIBUTE NUMBER field contains the attribute number (see 4.7.4) of the attribute associated with the error reported in the sense data.