To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 9 June 2006
Subject: 06-175r1 SPC-4 Wrapping and saturating counter definitions

Revision history

Revision 0 (27 March 2006) First revision

Revision 1 (9 June 2006) Included more context around some of the changes. After further discussion with Gerry Houlder (Seagate), did NOT change the Error Counter log pages to be allowed to contain wrapping counters as discussed in the May CAP WG meeting. 88-169r0, the proposal that originally defined the counters, was very clear that they are all saturating counters. The width of each counter is vendor-specific; if a counter is 64 bits, it will effectively never wrap or saturate (2^{64} = 18,446,744,073,709,551,616 events) no matter what it is counting. Since the new FORMAT AND LINKING field value of 10b allows data counters to continue incrementing when other data counters reach their maximum value, added wording to make it clear that this new mode is available in all the log pages containing data counters.

Related documents

spc4r04 - SCSI Primary Commands - 4 (SPC-4) revision 4
06-055r0 (aka 04-172r4) - SAS-2 More counters (Rob Elliott, HP)

Overview

SAS-2 includes definitions for wrapping and saturating counters that should also go into SPC-4.

Suggested changes to SPC-4

3.2.xxx saturating counter: A counter that remains at its maximum value after reaching its maximum value.

3.2.xxx wrapping counter: A counter that wraps back to zero after reaching its maximum value.

4.5.2.4.3 Actual retry count sense key specific data

...

The ACTUAL RETRY COUNT field returns vendor specific information on the number of retries of the recovery algorithm used in attempting to recover an error or exception condition.

NOTE 5 - This field should be computed in the same way as the retry count fields within the Read-Write Error Recovery mode page (see SBC-2, SSC-3, and MMC-4).

6.11 PERSISTENT RESERVE IN command

6.11.2 READ KEYS service action

...

The Persistent Reservations Generation (PRGENERATION) field shall contain a the value of a 32-bit wrapping counter maintained by the device server that shall be incremented every time a PERSISTENT RESERVE OUT command requests a REGISTER service action, a REGISTER AND IGNORE EXISTING KEY service action, a REGISTER AND MOVE service action, a CLEAR service action, a PREEMPT service action, or a PREEMPT AND ABORT service action. The counter shall not be incremented by a PERSISTENT RESERVE IN command, by a PERSISTENT RESERVE OUT command that performs a RESERVE or RELEASE service action, or by a PERSISTENT RESERVE OUT command that is terminated due to an error or reservation conflict. Regardless of the APTPL bit value, the PRgeneration value shall be set to zero by a power on.

7.2 Log parameters

7.2.1 Log page structure and page codes for all device types

...

Table 205 — Log page format

Byte\Bit	7	6	5	4	3	2	1	0
0	DS	SPF	PAGE CODE					
1	SUBPAGE CODE							
2	(MSB)	PAGE LENGTH (n-3)						
3								(LSB)
Log parameter(s)								
4	Log parameter (First)(see table 208) (Length x)							
x+3								
	. .							
n-y+1	Log parameter (Last)(see table 208) (Length y)							
n								

...

A FORMAT AND LINKING field set to 00b or 10b indicates that the parameter is a data counter. Data counters are saturating counters associated with one or more events. A data counter is ~~updated~~incremented whenever one of these events occurs~~ by incrementing the counter value~~. If ~~each~~a data counter has associated with it a vendor specific maximum value, then upon reaching this maximum value, the data counter shall not be incremented (i.e., its value does not wrap). When a data counter reaches its maximum value, the device server shall set the associated DU bit to one and handle other data counters in the log page as defined in table 210. If the data counter is at or reaches its maximum value during the processing of a command, the device server shall complete the command. If the command completes without error, except for the data counter being at its maximum value, and if the RLEC bit of the Control mode page (see 7.4.6) is set to one, then the device server shall terminate the command with CHECK CONDITION status, with the sense key set to RECOVERED ERROR, and the additional sense code set to LOG COUNTER AT MAXIMUM.

...

7.2.3 Buffer Over-Run/Under-Run log page

The Buffer Over-Run/Under-Run log page (page code 01h) defines 24 data counters that may be used to record the number of buffer over-runs or under-runs for the logical unit. A logical unit that implements this log page may implement one or more of the defined data counters.

A buffer over-run or under-run may occur when a SCSI initiator device does not transmit data to or from the logical unit's buffer fast enough to keep up with reading or writing the media. A buffer over-run condition may occur during a read operation when a buffer full condition prevents continued transfer of data from the media to the buffer. A buffer under-run condition may occur during a write operation when a buffer empty condition prevents continued transfer of data to the media from the buffer. Most devices incur a delay at this point while the media is repositioned.

The FORMAT AND LINKING field (see table 210 in 7.2.1) of each log parameter shall be set to 00b or 10b indicating it contains a data counter.

Table 216 defines the PARAMETER CODE field for the buffer over-run/under-run counters.

Table 216 — Parameter code field for buffer over-run/under-run counters

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	COUNT BASIS			CAUSE				TYPE

The PARAMETER CODE field for buffer over-run/under-run counters contains a 16-bit value comprised of eight reserved bits, a COUNT BASIS field (see table 217), a CAUSE field (see table 218), and a TYPE bit. These are concatenated to determine the value of the parameter code for that log parameter. (E.g., a counter for parameter code value of 0023h specifies a count basis of 001b, a cause of 0001b, and a type of 1b. This counter is incremented once per command that experiences an over-run due to the service delivery subsystem being busy.)

The COUNT BASIS field defines the criteria for incrementing the counter. The criteria are defined in table 217.

[Table 217 not shown]

The CAUSE field indicates the reason that the buffer over-run or under-run occurred. The following causes are defined in table 218.

[Table 218 not shown]

The TYPE bit indicates whether the counter records buffer under-runs or over-runs. A TYPE bit set to zero specifies a buffer under-run condition and a TYPE bit set to one specifies a buffer over-run condition.

~~The counters contain~~Each counter contains the total number of times buffer over-run or under-run conditions have occurred since the last time the counter was cleared. The counter shall be incremented for each occurrence of a~~n~~ buffer under-run or over-run condition and may be incremented more than once for multiple occurrences during the processing of a single command.

7.2.4 Error ~~c~~Counter log pages

This subclause defines the ~~e~~Error ~~c~~Counter log pages (see table 219).

Table 219 — Error ~~c~~Counter log page codes

Page Code	Log~~e~~ Page Name
03h	Read Error Counter
04h	Read Reverse Error Counter
05h	Verify Error Counter
02h	Write Error Counter

The log page format is defined in 7.2.1. A log page may return one or more log parameters that ~~record~~report data counters for events defined by the parameter codes. Table 220 defines the parameter codes for the ~~e~~Error ~~c~~Counter log pages.

Table 219 — Parameter codes for ~~e~~Error ~~c~~Counter log pages

Parameter code	Description
0000h	Errors corrected without substantial delay
0001h	Errors corrected with possible delays
0002h	Total (e.g., rewrites or rereads)
0003h	Total errors corrected
0004h	Total times correction algorithm processed
0005h	Total bytes processed
0006h	Total uncorrected errors
0007 - 7FFFh	Reserved
8000h - FFFFh	Vendor specific

NOTE 38 - The exact definition of the error counters is not part of this standard. These counters should not be used to compare products because the products may define errors differently.

The FORMAT AND LINKING field (see table 210 in 7.2.1) of each log parameter shall be set to 00b or 10b indicating it contains a data counter.

7.2.8 Non-Medium Error log page

The Non-Medium Error log page (page code 06h) provides for summing the occurrences of recoverable error events other than write, read, or verify failures. No discrimination among the various types of events is provided by parameter code (see table 225). Vendor specific discrimination may be provided through the vendor specific parameter codes.

Table 219 — Non-medium error event parameter codes

Parameter code	Description
0000h	Non-medium error count
0001h - 7FFFh	Reserved
8000h - FFFFh	Vendor specific error counts

The FORMAT AND LINKING field (see table 210 in 7.2.1) of each log parameter shall be set to 00b or 10b indicating it contains a data counter.

Editor's Note 1: please review the above assertion. Could they (particularly the vendor-specific error counts) contain list parameters as well?

7.2.11 Start-Stop Cycle Counter log page

...

The parameter value in the accumulated start-stop cycles log parameter (parameter code 0004h) shall contain a four-byte binary value that indicates how many stop-start cycles the SCSI target device has detected since its date of manufacture. The accumulated start-stop cycles counter is a saturating counter. If a LOG SELECT command attempts to change the value of the accumulated start-stop cycles log parameter, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST. The time at which the count is incremented during a start-stop cycle is vendor specific. For rotating magnetic storage devices, a single start-stop cycle is defined as an operational cycle that begins with the disk spindle at rest, continues while the disk accelerates to its normal operational rotational rate, continues during the entire period the disk is rotating, continues as the disk decelerates toward a resting state, and ends when the disk is no longer rotating. For devices without a spindle or with multiple spindles, the definition of a single start-stop cycle is vendor specific. The count is incremented by one for each complete start-stop cycle. No comparison with the value of parameter 0003h shall be performed by the device server. For the log parameter in which the parameter code value is 0004h, the values of the parameter control bits are defined in table 235.

7.3 Medium auxiliary memory attributes

7.3.2 Attribute identifier values

7.3.2.2 Device type attributes

7.3.2.2.2 LOAD COUNT: Indicates how many times this medium has been fully loaded. This attribute should not be reset to zero by any action of the device server. The load counter is a saturating counter.

7.3.2.2.4 INITIALIZATION COUNT: Indicates the number of times that a device server has logically formatted the medium. This value is cumulative over the life of the medium and shall not be reset to zero. The initialization counter is a saturating counter.

7.3.2.2.9 MEDIUM USAGE HISTORY: Provides saturating counters (see table 232) for the entire medium. The value in each field is the sum for all partitions. If a field is not used, it should be set to zero.

The CURRENT AMOUNT OF DATA WRITTEN field indicates the amount of data written to the medium during this load of the medium. This value is expressed in mebibytes (see 3.6.4).

The CURRENT WRITE RETRIES COUNT field indicates the total number of times a write retry occurred during this load of the medium.1)

The CURRENT AMOUNT OF DATA READ field indicates the amount of data read from the medium during this load of the medium. This value is expressed in mebibytes (see 3.6.4).

The CURRENT READ RETRIES COUNT field indicates the number of times a read retry occurred during this load of the medium.1)

The PREVIOUS AMOUNT OF DATA WRITTEN field indicates the amount of data written to the medium during the previous load of the medium. This value is expressed in mebibytes (see 3.6.4).

The PREVIOUS WRITE RETRIES COUNT field indicates the total number of times a write retry occurred during the previous load of the medium.1)

The PREVIOUS AMOUNT OF DATA READ field indicates the amount of data read from the medium during the previous load of the medium. This value is expressed in mebibytes (see 3.6.4).

The PREVIOUS READ RETRIES COUNT field indicates the number of times a read retry occurred during the previous load of the medium.1)

The TOTAL AMOUNT OF DATA WRITTEN field indicates the amount of data written to the medium since the last medium format. This value is expressed in mebibytes (see 3.6.4).

The TOTAL WRITE RETRIES COUNT field indicates the total number of times a write retry occurred since the last medium format.1)

The TOTAL AMOUNT OF DATA READ field indicates the amount of data read from the medium since the last medium format. This value is expressed in mebibytes (see 3.6.4).

The TOTAL READ RETRIES COUNT field indicates the number of times a read retry occurred since the last medium format.1)

The LOAD COUNT field indicates the number of loads since the last medium format. This count accumulates over the life of the medium but it is reset to zero after a medium format.

The TOTAL CHANGE PARTITION COUNT field indicates the number of times that switches between partitions have been performed on the medium. This count accumulates over the life of the medium but it is reset to zero after a medium format.

The TOTAL PARTITION INITIALIZE COUNT field indicates number of times that any of the partitions on the medium have been erased. This count accumulates over the life of the medium but it is reset to zero after a medium format.

...

Footnote 1) The definition of one retry as counted by this attribute field is not part of this standard. This count~~er~~ should not be used to compare products because the products may define errors differently.

7.3.2.2.10 PARTITION USAGE HISTORY: Provides saturating counters (see table 233) for the partition specified by the PARTITION NUMBER field in the CDB. If a field is not used, it should be set to zero.

The CURRENT AMOUNT OF DATA WRITTEN field indicates the amount of data written to the medium in the partition specified by the PARTITION NUMBER field in the CDB during this load of the medium. This value is expressed in mebibytes (see 3.6.4).

The CURRENT WRITE RETRIES COUNT field indicates the total number of times a write retry occurred in the partition specified by the PARTITION NUMBER field in the CDB during this load of the medium. 2)

The CURRENT AMOUNT OF DATA READ field indicates the amount of data read from the medium in the partition specified by the PARTITION NUMBER field in the CDB during this load of the medium. This value is expressed mebibytes (see 3.6.4).

The CURRENT READ RETRIES COUNT field indicates the number of times a read retry occurred in the partition specified by the PARTITION NUMBER field in the CDB during this load of the medium.2)

The PREVIOUS AMOUNT OF DATA WRITTEN field indicates the amount of data written to the medium in the partition

specified by the PARTITION NUMBER field in the CDB during the previous load of the medium. This value is expressed in mebibytes (see 3.6.4).

The PREVIOUS WRITE RETRIES COUNT field indicates the total number of times a write retry occurred in the partition specified by the PARTITION NUMBER field in the CDB during the previous load of the medium.2)

The PREVIOUS AMOUNT OF DATA READ field indicates the amount of data read from the medium in the partition

specified by the PARTITION NUMBER field in the CDB during the previous load of the medium. This value is expressed in mebibytes (see 3.6.4).

The PREVIOUS READ RETRIES COUNT field indicates the number of times a read retry occurred in the partition specified by the PARTITION NUMBER field in the CDB during the previous load of the medium.2)

The TOTAL AMOUNT OF DATA WRITTEN field indicates the amount of data written to the medium in the partition specified by the PARTITION NUMBER field in the CDB since the last medium format. This value is expressed in mebibytes (see 3.6.4).

The TOTAL WRITE RETRIES COUNT field indicates the total number of times a write retry occurred in the partition specified by the PARTITION NUMBER field in the CDB since the last medium format.2)

The TOTAL AMOUNT OF DATA READ field indicates the amount of data read from the medium in the partition specified by the PARTITION NUMBER field in the CDB since the last medium format. This value is expressed in mebibytes (see 3.6.4).

The TOTAL READ RETRIES COUNT field indicates the number of times a read retry occurred in the partition specified by the PARTITION NUMBER field in the CDB since the last medium format.2)

The LOAD COUNT field indicates the number of loads in the partition specified by the PARTITION NUMBER field in the CDB since the last medium format. This count accumulates over the life of the medium but it is reset to zero after a medium format.

The TOTAL CHANGE PARTITION COUNT field indicates the number of times that switches to the partition specified by the PARTITION NUMBER field in the CDB have been performed on the medium. This count accumulates over the life of the medium but it is reset to zero after a medium format.

The TOTAL PARTITION INITIALIZE COUNT field indicates number of times that the partition specified by the PARTITION NUMBER field in the CDB has been initialized. This count accumulates over the life of the medium but it is reset to zero after a medium format.

Footnote 2) The definition of one retry as counted by this attribute field is not part of this standard. This count~~er~~ should not be used to compare products because the products may define errors differently.

7.4 Mode parameters

7.4.11 Informational Exceptions Control mode page

...

The ~~value in the~~ INTERVAL TIMER field ~~is~~specifies the period in 100 millisecond increments that the device server shall use for reporting that an informational exception condition has occurred. The device server shall not report informational exception conditions more frequently than the time specified by the INTERVAL TIMER field and shall report them after the time specified by INTERVAL TIMER field has elapsed. After the informational exception condition has been reported the interval timer shall be restarted. An ~~value of zero or FFFF FFFFh in the~~ INTERVAL TIMER field ~~indicates~~set to zero or FFFF FFFFh specifies that the period for reporting an informational exception condition is vendor specific.

The ~~value in the~~ REPORT COUNT field ~~is~~specifies the maximum number of times the device server may ~~to~~ report an informational exception condition to the application client. A ~~value of zero in the~~ REPORT COUNT field ~~indicates~~set to zero specifies that there is no limit on the number of times the device server may report~~s~~ an informational exception condition.

The maintaining of the interval timer and the report counter across power cycles, hard resets, logical unit resets, and I_T nexus losses is vendor specific.

Editor's Note 2: since the report counter is not accessible, there is no need to mention that the report counter is saturating

8.3 ACCESS CONTROLS well known logical unit

8.3.1.4 Managing the ACL

8.3.1.4.3 Identifying logical units during ACL management

...

The association between default LUN values and logical units is managed by the access controls coordinator and may change due to circumstances that are beyond the scope of this standard. To track such changes, the access controls coordinator ~~shall~~ maintains a ~~generation counter value called~~ DLgeneration counter as described in 8.3.1.4.4.

8.3.1.4.4 Tracking changes in logical unit identification

The access controls coordinator shall ~~maintain a generation counter value called~~implement a Default LUNs Generation (DLgeneration) counter to track changes in the association between default LUN values and logical units. The DLgeneration counter is a wrapping counter.

When access controls are disabled the DLgeneration counter shall be set to zero. When access controls are first enabled (see 8.3.1.2) the DLgeneration counter shall be set to one. While access controls are enabled, the access controls coordinator shall increment the DLgeneration counter by one every time the association between default LUN values and logical units changes (e.g., following the creation of a new logical unit, deletion of an existing logical unit, or removal and recreation of an existing logical unit).

The access controls coordinator shall include the current DLgeneration value in the parameter data returned by an ACCESS CONTROL IN command with REPORT LU DESCRIPTORS (see 8.3.2.3) or REPORT ACL (see 8.3.2.2) service action. The application client shall supply the DLgeneration value for the default LUN values it is using in the parameter data for an ACCESS CONTROL OUT command with MANAGE ACL service action (see 8.3.3.2).

Before processing the ACL change information in the parameter list provided by an ACCESS CONTROL OUT command with MANAGE ACL service action, the access controls coordinator shall verify that the DLgeneration value in the parameter data matches the current value of the DLgeneration counter ~~currently in use~~. If the DLgeneration value verification finds a mismatch, the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

8.3.1.10 Access controls log

The access controls log is a record of events maintained by the access controls coordinator.

The access controls log has three portions, each recording a different class of events:

a) **invalid key events**: mismatches between the management identifier key (see 8.3.1.8) specified by a service action and the current value maintained by the access controls coordinator;
b) **key override events**: attempts to override the management identifier key (see 8.3.1.8.2.1), whether the attempt fails or succeeds; and
c) **ACL LUN conflict events**: (see 8.3.1.5.2).

Each portion of the log is required to contain a saturating counter of the corresponding events. When a SCSI target device is manufactured, all event counters shall be set to zero. When access controls are disabled, all counters except the ~~k~~Key ~~o~~Override events counter shall be set to zero. Each event counter shall be incremented by one whenever the relevant event occurs.

Each log portion may contain additional records with more specific information about each event. When the resources for additional log records are exhausted, the access controls coordinator shall preserve the most recently added log records in preference to older log records.

Log records contain a TIME STAMP field whose contents are vendor specific. If the access controls coordinator has no time stamp resources the TIME STAMP field shall be set to zero. If time stamp values are provided, the same timing clock and time stamp format shall be used for all access controls log entries.

Invalid key events occur whenever an access controls command requires the checking of an application client supplied management identifier key against the current management identifier key saved by the access controls coordinator and the two values fail to match. When such an event occurs, the access controls coordinator shall increment the ~~i~~Invalid ~~k~~Key~~s~~ event counter by one. If the log has additional resources to record event details, the access controls coordinator shall add an invalid keys log record (containing the information defined in 8.3.2.4.2.3) describing the event.

Key override events occur when the access controls coordinator receives the ACCESS CONTROL OUT command with OVERRIDE MGMT KEY service action (see 8.3.3.8). When such an event occurs, the access controls coordinator shall increment the ~~k~~Key ~~o~~Override~~s~~ event counter by one without regard for whether the command succeeds or fails. If the log has additional resources to record event details, the access controls coordinator shall add a~~n~~ key overrides log record (containing the information defined in 8.3.2.4.2.2) describing the event.

ACL LUN conflict events occur as specified in 8.3.1.5.2. When such an event occurs, the access controls coordinator shall increment the ACL LUN ~~c~~Conflict~~s~~ event counter by one. If the log has additional resources to record event details, the access controls coordinator shall add an ACL LUN conflicts log record (containing the information defined in 8.3.2.4.2.4) describing the event.

Selected portions of the access controls log may be requested by an application client using the ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action (see 8.3.2.4). With the exception of the key overrides portion, selected portions of the log may be cleared and the event counters reset to zero using the ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action (see 8.3.3.6).

8.3.1.12 Access controls information persistence and memory usage requirements

...

Table 351 — Mandatory access controls resources

Information description	Size	...
Access Controls Log Event Counters (see 8.3.1.10) containing at least the following: a) Key Override~~s~~ ~~Counter~~event counter; b) Invalid Key~~s~~ ~~Counter~~event counter; and c) ACL LUN Conflict~~s~~ ~~Counter~~event counter	...	...
...	...	...

..

Table 352 — Optional access controls resources

Information description	Size	...
Access Controls log event records (see 8.3.1.10) for: a) Key Override~~s~~ events; b) Invalid Key~~s~~ events; and c) ACL LUN Conflict~~s~~ events	...	...
...	...	...

...

8.3.2 ACCESS CONTROL IN command

8.3.2.2.2 REPORT ACL parameter data introduction

...

The DLGENERATION field shall contain the current DLgeneration value (see 8.3.1.4.4).

8.3.2.2.2.2 Granted ACL data page format

...

The DEFAULT LUN field identifies the logical unit to which access is allowed using the default LUN value described in 8.3.1.4.3. The value in the DEFAULT LUN field shall be consistent with the DLGENERATION field ~~contents~~value returned in the parameter list header (see 8.3.2.2.2).

8.3.2.2.2.4 Proxy Tokens ACL data page format

...

The DEFAULT LUN field identifies the logical unit to which this proxy token allows access using the default LUN value described in 8.3.1.4.3. The value in the DEFAULT LUN field shall be consistent with the DLGENERATION field value returned in the parameter list header (see 8.3.2.2.2).

8.3.2.3.2 REPORT LU DESCRIPTORS parameter data format

...

The DLGENERATION field shall contain the current DLgeneration value (see 8.3.1.4.4).

...

The DEFAULT LUN field contains the default LUN value (see 8.3.1.4.3) for the logical unit described by this logical unit descriptor. The value in the DEFAULT LUN field shall be consistent with the DLGENERATION field value returned in the parameter list header (see 8.3.2.3.2). The value in the DEFAULT LUN field shall not identify a well known logical unit.

8.3.2.4 REPORT ACCESS CONTROLS LOG service action

8.3.2.4.2.1 REPORT ACCESS CONTROLS LOG parameter data introduction

...

The COUNTER field contains the event~~s~~ counter value (see 8.3.1.10) for the access controls log portion indicated by the LOG PORTION field (see table 372).

Table 372 — Parameter data LOG PORTION field values

...	...	COUNTER field contents	...
...	...	Key Override~~s~~ event counter Invalid Key~~s~~ event counter ACL LUN Conflict~~s~~ event counter	...

8.3.2.4.2.2 Key Overrides access controls log portion page format

...

The INITIAL OVERRIDE LOCKOUT TIMER field shall contain the access controls coordinator's initial override lockout timer value (see 8.3.1.8.2.2) at the time when the key override event was logged.

The OVERRIDE LOCKOUT TIMER field shall contain the access controls coordinator's override lockout timer value (see 8.3.1.8.2.2) at the time when the key override event was logged.

8.3.2.5 REPORT OVERRIDE LOCKOUT TIMER service action

...

The KEY OVERRIDES COUNTER field shall be set to the value of the ~~k~~Key ~~o~~Override~~s~~ event counter in the access controls log (see 8.3.1.10).

8.3.3 ACCESS CONTROL OUT command

8.3.3.2.1 MANAGE ACL introduction

...

The DLGENERATION field specifies the DLgeneration value (see 8.3.1.4.4) associated with the default LUN values in the Grant/Revoke ACE pages in the parameter data.

8.3.3.2.2 The Grant/Revoke ACE page

...

a) The ~~contents~~value of the DLGENERATION field in the parameter list header (see 8.3.3.2.1) does not match the current value of the DLgeneration counter ~~value~~ (see 8.3.1.4.4) maintained by the access controls coordinator;

...

The DEFAULT LUN field specifies the logical unit to which the value in the LUN VALUE allows access. The DEFAULT LUN field shall contain a default LUN value (see 8.3.1.4.3). The value in the DEFAULT LUN field shall be consistent with the DLGENERATION field ~~contents~~value specified in the parameter list header (see 8.3.3.2.1). If the DEFAULT LUN field references a well known logical unit, the access controls coordinator's state shall not be modified and the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.

8.3.3.2.3 The Grant All ACE page

...

c) A DEFAULT LUN field whose contents reference the logical unit appropriate to the DLgeneration value (see 8.3.1.4.4).

8.3.3.3 DISABLE ACCESS CONTROLS service action

...

In response to an ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS service action with correct management identifier key value the access controls coordinator shall:

a) Disable access controls;
b) Clear the ACL (see 8.3.1.3);
c) Place all initiator ports into the not-enrolled state (see 8.3.1.5.1);
d) Set the management identifier key to zero (see 8.3.1.8);
e) Set the override lockout timer to zero (see 8.3.1.8.2.2);
f) Set the initial override lockout timer value to zero (see 8.3.1.8.2.2);
g) Clear the access controls log, including resetting event counters to zero, with the exception of the key overrides portion of the access controls log (see 8.3.1.10);
h) Allow all initiator port's access to all logical units at their default LUN value;
i) Optionally, ~~re~~set the DLgeneration ~~value~~counter to zero (see 8.3.1.4.4); and
j) Establish a unit attention condition for the initiator port associated with every I_T nexus, with the additional sense code set to REPORTED LUNS DATA HAS CHANGED.

...

8.3.3.6 CLEAR ACCESS CONTROLS LOG service action

The ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action is used to instruct the access controls coordinator to reset a specific access control ~~log~~event counter to zero and to clear a portion of the access controls log (see 8.3.1.10). If the ACCESS CONTROL OUT command is implemented, the CLEAR ACCESS CONTROLS LOG service action shall be implemented.

...

In response to an ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action with correct management identifier key value the access controls coordinator shall perform the following to clear the portion of the access controls log identified by the LOG PORTION field (see table 393) in the parameter data:

a) Set the event counter for the specified log portion to zero; and
b) If the specified log portion contains log records, remove the log records from the specified log portion.

Annex C Procedures for logging operations in SCSI

C.2.7 parameter value: A data counter, cumulative, threshold, or ASCII value.

C.5.3 Pseudocode 2

IF a ~~log~~ data counter reaches its maximum value:

```
1) Set the DU bit to 1, indicating that the device server is no longer updating the log parameter
2) IF there is no active task
    1) Wait until there is an active task
    END
3) Process the command in the active task
4) Complete the command with CHECK CONDITION status, with the sense key set to RECOVERED
ERROR and the additional sense code set to LOG EXCEPTION, COUNT AT MAXIMUM
5) IF the cause of the counter reaching maximum is not cleared by the application client
    1) Do not return CHECK CONDITION status and do not increment the counter
    END
END
```