ENDL
TEXAS

Date: 4 September 2001
To: T10 Technical Committee
From: Ralph O. Weber & Jim Hafner
Subject: Access Controls for SPC-3 (the rewrite)

## A – Introduction

The access controls proposal has been modified by several additional documents, producing the following list of proposals all related to access controls:

|  |  |
|---|---|
| 99-245r9 | A Detailed Proposal For Access Controls |
| 00-261r0 | Discussion of editorial changes to Access Controls in 99-245r9 |
| 00-287r1 | TransportIDs for Access Controls |
| 00-381r0 | Three minor modifications to Access Controls in SPC-3 |
| 01-026r1 | SPC-3 Access Controls LUN conflicts due to transport IDs |

In addition, past experience with complex proposals such as persistent reservations suggests that a rewrite of the original proposal by someone with T10 editing experience will occur sooner or later. Why not sooner?

This proposal intends to make no technical changes to the approved access controls proposals listed above. However, some omissions have been discovered during the rewrite process and statements have been added that may constitute requirements not previously noted explicitly. The author believes that the original intent has been maintained throughout.

Familiarity with the access controls concepts is assumed. This proposal contains almost none of the explanatory text found in the proposals listed above and reference is made to those proposals for the historical perspective.

It is anticipate that a few revisions will be needed before this proposal can be approved. While that work is in progress, it is recommended that incorporation of the proposals listed above be deferred in SPC-3. Once completed, this proposal should replace all of them.

## B – Notations Used

Editorial additions are indicated with blue text and editorial deletions are indicated with blue strike through text. When small blocks of text is moved verbatim from one location to another, the removal site has the text in green strike through and the insertion site has the text in green. When whole clauses are moved verbatim from one location to another, the removal site has the text in yellow with no strike through and the insertion site has the text in green.

The intent is to remove the yellow and green markings as soon as possible, perhaps before r0 is published.

Red text that is double underlined indicates where new statements have been added that the author believes are statements of previously unstated requirements.

## C – Issues to Work

### C.1 – FCP-2 Glossary

The glossary entries as added in FCP-2 rev 7 for access controls are not exactly as given here. They are:

**access controls**: Mechanisms allowing a managing application client to control the set of initiators that have access to a target. The access control is enforced by the target. See SPC-3.

**access controls data**: Information sent to the target by the managing application client that is used by the target to control the set of initiators that have access to the target. See SPC-3.

**access controls enrollment state**: A state established in the target by the managing application client. This state governs the behavior of the target in controlling the set of initiators that have access to the target. See SPC-3.

## D – Significant Changes to the Approved Access Controls Proposal

In the course of preparing this rewrite of the access controls proposal, a couple of issues have surfaced that I think should be considered as significant changes to the proposal:

a) I LUN 0 access model described in the approved proposal should be removed in favor of accessing the access controls coordinator via a well known logical unit. The effects of this change are:
   A) The access controls model, commands, and data go in clauses separate from the 'all device types' clauses
   B) The access controls coordinator will need its own list of supported commands, probably only INQUIRY, ACCESS CONTROL IN, and ACCESS CONTROL OUT;
   C) Discussion on LUN 0 usage is removed from the ACCESS CONTROL IN and ACCESS CONTROL OUT; and
   D) The ACC bit in the standard INQUIRY data would be either left reserved or made vendor specific based on the status of existing implementations; also

b) I believe that a new unit attention condition needs to be defined into which all logical units in a SCSI target device are placed after access controls are disabled. In the absence of such a unit attention condition, initiators may continue operating under obsolete understandings of LUN to logical unit relationships following the disabling of access controls.

## E – Summary Information

The following information summarizes proposed code values. It is not directly part of this proposal.

### E.1 – Access Control Operation Codes and Service Actions

Table 1 summarizes the service actions for the ACESS CONTROL IN command (operation code 86h).

Table 1: ACESS CONTROL IN Service Actions

| Code | Name | Type | Clause |
|---|---|---|---|
| 00h | REPORT ACL | M | 7.1.2 |
| 01h | REPORT LU DESCRIPTORS | M | 7.1.3 |
| 02h | REPORT ACCESS CONTROLS LOG | M | 7.1.4 |
| 03h | REPORT OVERRIDE LOCKOUT TIMER | M | 7.1.5 |
| 04h | REQUEST PROXY TOKEN | O | 7.1.6 |
| 05h-17h | Reserved | | |
| 18h-1Fh | Vendor-specific | V | |

Table 2 summarizes the service actions for the ACESS CONTROL OUT command (operation code 87h).

Table 2: ACESS CONTROL OUT Service Actions

| Code | Name | Type | Clause |
|---|---|---|---|
| 00h | MANAGE ACL | M | 7.2.2 |
| 01h | DISABLE ACCESS CONTROLS | M | 7.2.3 |
| 02h | ACCESS ID ENROLL | M | 7.2.4 |
| 03h | CANCEL ENROLLMENT | M | 7.2.5 |
| 04h | CLEAR ACCESS CONTROLS LOG | M | 7.2.6 |
| 05h | MANAGE OVERRIDE LOCKOUT TIMER | M | 7.2.7 |
| 06h | OVERRIDE MGMT ID KEY | M | 7.2.8 |
| 07h | REVOKE PROXY TOKEN | O | 7.2.9 |
| 08h | REVOKE ALL PROXY TOKENS | O | 7.2.10 |
| 09h | ASSIGN PROXY LUN | O | 7.2.11 |
| 0Ah | RELEASE PROXY LUN | O | 7.2.12 |
| 0Bh-17h | Reserved | | |
| 18h-1Fh | Vendor-specific | V | |

## E.2 – Access Control Additional Sense Codes

Table 3 contains a list of the Additional Sense Code and Additional Sense Code Qualifiers relevant to access controls. Section F.5 formally proposes the addition of these codes. ~~The contents of this table, suitably modified for inclusion in SPC-3, may be found in Appendix C (Table 42).~~

Table 3: Access Control Additional Sense Codes and Qualifiers

| ASC | ASCQ | Description | Description |
|---|---|---|---|
| 20h | 01h | ACCESS DENIED - INITIATOR PENDING-ENROLLED | An initiator in the pending-enrolled state sends a restricted command to a logical unit accessible under the enrolled AccessID. |
| 20h | 02h | ACCESS DENIED - NO ACCESS RIGHTS | An initiator in the not-enrolled state sends an ACCESS ID ENROLL service action and the given AccessID has no access rights in the ACL. |
| 20h | 03h | ACCESS DENIED - INVALID MGMT ID KEY | The Management Identifier Key value does not match the value maintained by the access controls coordinator. |
| 20h | 08h | ACCESS DENIED - ENROLLMENT CONFLICT | An initiator in the enrolled or pending-enrolled state issues an ACCESS ID ENROLL service action under a different AccessID. |
| 20h | 09h | ACCESS DENIED - INVALID LU IDENTIFIER | A LUN or default LUN value in a CDB field or parameter data is not valid. |
| 20h | 0Ah | ACCESS DENIED - INVALID PROXY TOKEN | The Proxy Token is not valid; it does not correspond to a logical unit. |
| 20h | 0Bh | ACCESS DENIED - ACL CONFLICT | The enrollment failed because an ACL conflict occurred. |
| 55h | 05h | INSUFFICIENT ACCESS CONTROL RESOURCES | The access controls coordinator has exhausted its resources for the requested access controls action. |

# F – Changes Proposed for SPC-3

## F.1 – Glossary and Acronyms

The following additions to the glossary and acronyms clause of SPC-3 are proposed.

### F.1.1 – Glossary

**3.1.r access controls:** An optional SCSI target device feature that restricts initiator access to specific logical units and modifies the information about logical units in the parameter data of INQUIRY and REPORT LUNS commands (see 5.99).

**3.1.s access control list (ACL):** The data used by a SCSI target device to configure access rights for initiators according to the access controls state of the SCSI target device (see 5.99.2).

**3.1.t access control list entry (ACE):** One entry in the access control list (see 3.1.s).

**3.1.u access controls coordinator:** The entity within a SCSI target device that coordinates the management and enforcement of access controls (see 5.99) for all logical units within the SCSI target device. The access controls coordinator is always addressable through LUN 0.

**3.1.v logical unit access control descriptor (LUACD):** The structure within an ACE (see 3.1.t) that identifies a logical unit to which access is allowed and specifies the LUN by which the logical unit is to be accessed (see 5.99.2.3).

**3.1.w proxy token:** An identifier for a logical unit that may be used to gain temporary access to that logical unit in the presence of access controls (see 5.99.5.2).

### F.1.2 – Acronyms

| | |
|---|---|
| **ACE** | **Access Control list Entry (see 3.1.t)** |
| **ACL** | Access Control List (see 3.1.s) |
| LUACD | Logical Unit Access Control Descriptor (see 3.1.v) |
| SBC-2 | SCSI Block Commands -2 (see clause 1) |

## F.2 – Access Controls & Reservations

Table 4 shows two lines to be added in SPC-3 for the new commands introduced by access controls. In SPC-3 revision 00, the affected table was table 10.

Table 4: SPC-3 Reservations Conflicts Table Changes for Access Controls

| Command | Addressed LU is reserved by another initiator [A] | Addressed LU has this type of persistent reservation held by another initiator [B] | | | | |
|---|---|---|---|---|---|---|
| | | From any initiator | | From registered initiator (RO all types) | From initiator not registered | |
| | | Write Excl | Excl Access | | Write Excl RO | Excl Access – RO |
| ACCESS CONTROL IN | Allowed | Allowed | Allowed | Allowed | Allowed | Allowed |
| ACCESS CONTROL OUT | Allowed | Allowed | Allowed | Allowed | Allowed | Allowed |

## F.3 – Changes to the EXTENDED COPY command

In the target descriptor formats in SPC-Tables 17, 19, 20, 21, 22, and 23, change byte3, bits 0-1 to a new 2-bit field called LU ID TYPE. In SPC-Table 19, 20, 21, and 22, change the LOGICAL UNIT NUMBER field name to LU IDENTIFIER.

Add the following paragraphs to clause SPC-7.2.6.1 after the paragraph that begins “The copy manager may,...”:

The LU ID TYPE field (see table 1) specifies ~~determines~~ the interpretation of the LU IDENTIFIER field in ~~some~~ target descriptors that contain a LU IDENTIFIER field ~~(see SPC-7.2.6.2, SPC-7.2.6.3, SPC-7.2.6.4, and SPC-7.2.6.5)~~.

**Table t1 — LU ID type codes**

| Type Code | LU IDENTIFIER **field contents** | Reference |
|---|---|---|
| 00b | Logical Unit Number | SAM-2 |
| 01b | Proxy Token | 5.99.5.2 |
| 10b - 11b | Reserved | |

~~This is described in Table 5. In all other target descriptors this field is reserved.~~

~~TABLE 5: LU ID TYPE and LU IDENTIFIER description~~

| ~~LU ID Type~~ | ~~LU Identifier description~~ |
|---|---|
| ~~00b~~ | ~~Logical Unit Number~~ |
| ~~01b~~ | ~~Proxy Token~~ |
| ~~10b-11b~~ | ~~Reserved~~ |

Support for LU ID type codes ~~values~~ other than 00b ~~(Logical Unit Number)~~ is optional. If a copy manager receives an unsupported LU ID type code ~~value in a target descriptor~~, the command shall be terminated with a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

If the LU ID TYPE field specifies ~~indicates~~ that the LU IDENTIFIER field contains a logical unit number, then the LU IDENTIFIER field specifies the logical unit within the SCSI device specified ~~addressed~~ by other fields in the target descriptor ~~(as defined in each subclause)~~ that shall be the ~~target (~~source or destination~~)~~ for EXTENDED COPY operations.

If the LU ID TYPE field specifies ~~indicates~~ that the LU IDENTIFIER field contains a proxy token (see 5.99.5.2), then the copy manager shall use the LU IDENTIFIER field contents to obtain ~~specifies an access controls Proxy Token (see 0.0.19) that shall be used by the copy manager to gain~~ proxy access rights to the ~~relevant~~ logical unit associated with the proxy token. The logical unit number that represents the proxy access rights ~~that~~ shall be the ~~target (~~source or destination~~)~~ for EXTENDED COPY operations.

The copy manager should obtain a LUN value for addressing this logical unit by sending ~~the Proxy Token in parameter data for the~~ an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 7.2.11) to the access controls coordinator of the SCSI device that is identified by other fields in the target descriptor. The copy manager shall ~~send to~~ use ~~the~~ a LUN assigned on the basis of ~~this~~ a proxy token only for those commands that are necessary for the processing ~~completion~~ of ~~those~~ the EXTENDED COPY ~~commands that contain this~~ command whose parameter data contains the proxy token ~~value in their target descriptors~~. When the copy manager has completed EXTENDED COPY commands involving ~~that~~ a proxy token, the copy manager should release the LUN value using an ~~by sending the~~ ACCESS CONTROL OUT command with RELEASE PROXY LUN service action (see 7.2.12) ~~to the access controls coordinator of the SCSI device identified in the target descriptor~~.

EXTENDED COPY access to proxy logical units is to be accomplished only via LU ID type 01b. If the copy manager receives a target descriptor containing LU ID type 00b and a logical unit number matching a LUN value that the copy manager has obtained using an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action, the EXTENDED COPY command shall be terminated with a ~~that specifies a logical unit either by LUN or by logical unit identifier and the copy manager has access to that logical only under the rights of one or more Proxy Tokens, it shall reject the command with~~ CHECK CONDITION status, the sense key shall be set to COPY ABORTED and the additional sense code shall be set to COPY TARGET DEVICE NOT REACHABLE.

In each subclause SPC-7.2.6.2-7.2.6.5, remove the paragraph which starts “The LOGICAL UNIT NUMBER...” and replace it with the following paragraph:

The LU ID TYPE field and LU IDENTIFIER field are described in SPC-7.2.6.1.

In the subclause SPC-7.2.6.6, insert the following paragraph after the paragraph which starts “The contents of...”

The LU ID TYPE field is reserved for this target descriptor.

## F.4 – Changes to the Standard INQUIRY Data

In the standard INQUIRY data format, SPC-Table 46, make the following change. Byte 5, bit 6 is changed from Reserved to ACC (for Access Controls Coordinator). The following additional text be added after the paragraph describing the SCCS bit in clause SPC-7.3.2:

An Access Controls Coordinator (ACC) bit of one indicates that the device contains an access controls coordinator that may be addressed through this logical unit. An ACC bit of zero indicates that no access controls coordinator is present. If the device contains an access controls coordinator, the ACC bit shall be set to one for LUN 0.

## F.5 – New additional sense codes

The following ASC/ASCQ codes should be added and marked as used by all device types:

| | |
|---|---|
| 20h/01h | ACCESS DENIED - INITIATOR PENDING-ENROLLED |
| 20h/02h | ACCESS DENIED - NO ACCESS RIGHTS |
| 20h/03h | ACCESS DENIED - INVALID MGMT ID KEY |
| 20h/08h | ACCESS DENIED - ENROLLMENT CONFLICT |
| 20h/09h | ACCESS DENIED - INVALID LU IDENTIFIER |
| 20h/0Ah | ACCESS DENIED - INVALID PROXY TOKEN |
| 20h/0Bh | ACCESS DENIED - ACL LUN CONFLICT |
| 55h/05h | INSUFFICIENT ACCESS CONTROL RESOURCES |

## F.6 – Access Controls Model

This is the model clause for addition to SPC-3.

## 5.99 Access Controls

### 5.99.1 Access controls overview

Access controls are an optional SCSI target device feature that application clients may use to restrict logical unit access to specified initiators or groups of initiators ~~to allow only specified initiators or groups of initiators to access specified logical units~~.

Access controls are handled in the SCSI target device ~~at the target~~ by an access controls coordinator. The access controls coordinator associates a specific LUN to a specific logical unit depending on which initiator accesses the SCSI target device and whether the initiator has rights to the logical unit.

Access rights to a logical unit affects whether the logical unit appears in the parameter data returned by a REPORT LUNS command and how the logical unit responds to INQUIRY commands.

The access controls coordinator maintains information about which initiators are allowed access to which logical units via which LUNs in the access control list (ACL), described in 5.99.2. The format of the ACL is vendor specific.

To support third party commands such as EXTENDED COPY, the access controls coordinator may provide proxy tokens (see 5.99.5.2) to allow one initiator to pass its access capabilities to another initiator.

An application client ~~may~~ manages the access controls state of the SCSI target device using ~~access control~~ the following commands:

a) ACCESS CONTROL IN - ~~queries~~ to request information or a proxy token from the access controls coordinator ~~information~~; and
b) ACCESS CONTROL OUT
   A) to allow ~~creates~~, change~~s~~ or revoke~~s~~ logical unit access ~~controls~~;
   B) to revoke a proxy token; and
   C) otherwise to manage~~s~~ the access controls coordinator.

~~The access control commands are not subject to reservation conflicts.~~

A SCSI device has access controls disabled when it is shipped from the factory and after successful completion of the ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS service action. In this state, the ACL ~~is empty (has~~ contains no entries~~)~~ and the management identifier key (see 5.99.7) is zero.

The first successful ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2) shall enable~~s~~ access controls~~, that is, transitions the device to the **access controls enabled** state~~. ~~In this state~~ When access controls are enabled, all logical units ~~are~~ shall be inaccessible to all initiators unless the ACL (see 5.99.2) allows access ~~there are specific rights granted to specific initiators as defined in the ACL~~.

The ACL allows an initiator ~~has~~ access to a logical unit if the ACL contains an ACE (see 5.99.2) with an access identifier (see 5.99.2.2) associated with the initiator and that ACE contains a LUACD (see 5.99.2.3) that references the logical unit. ~~as specified in the ACL if that initiator is identified by an access identifier (see x.x.x) in an ACL entry and that logical unit is referenced in an accessible logical unit pair in that ACL entry. In this case, the LUN value for this logical unit as would be returned in parameter data for a REPORT LUNS command from this initiator shall be the LUN value of the accessible logical unit pair in that ACL entry.~~ When the ACL allows access to a logical unit, the REPORT LUNS command parameter data bytes representing that logical unit shall contain the LUN value found in the LUACD that references that logical unit and the initiator shall use the same LUN value when sending commands to the logical unit.

A~~dditionally, a~~n initiator also may ~~gain~~ be allowed access to a logical unit ~~with~~ through the use of a proxy token (see 5.99.5.2).

Once access controls are enabled, they shall remain enabled until:

a) successful completion of an ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS service action; or
b) vendor specific physical intervention.

Successful downloading of firmware may result in access controls being disabled.

Once access controls are enabled, power cycles, logical unit resets, and target resets shall not disable them.

### 5.99.2 The access control list

#### 5.99.2.1 ACL overview

The specific access controls ~~of the~~ for a SCSI target device are instantiated by the access controls coordinator using data in an access control~~s~~ list (ACL). The ACL ~~consists of~~ contains zero or more access control list entries (ACEs), each ~~entry containing~~ ACE contains the following:

a) one access identifier (see 5.99.2.2) that identifies the initiator(s) to which the ACE applies; and
b) a list of ~~accessible logical unit pairs, each pair consisting of one LUN value and a reference to one logical unit.~~ logical unit access control descriptors (LUACDs) that identify the logical units to which the initiator(s) have access and the LUNs used to access those logical units by the initiator(s), each LUACD (see 5.99.2.3) contains the following:
   A) a vendor specific reference for the logical unit; and
   B) a LUN.

Figure f1 shows the structure of an ACL.

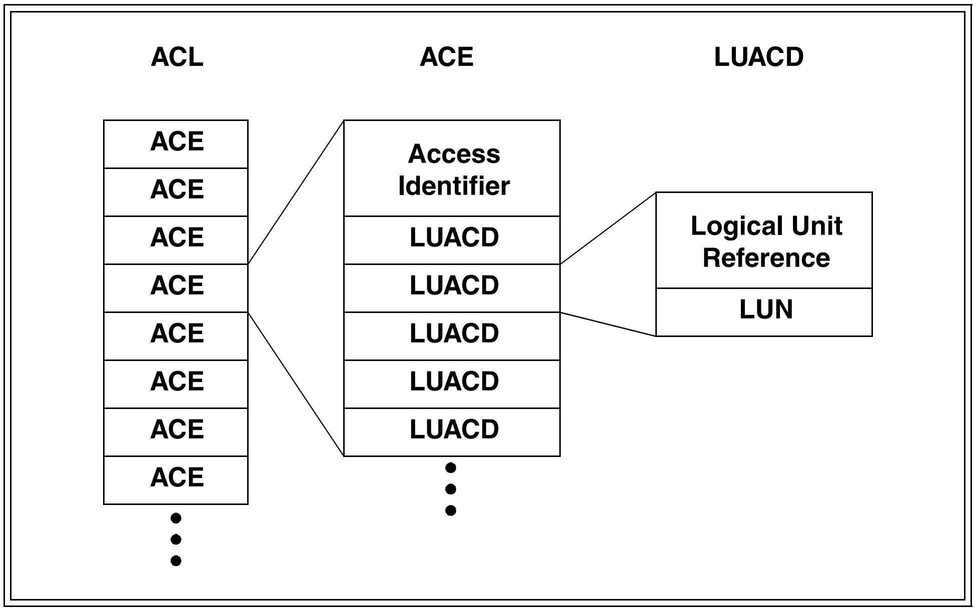

**Figure f1 — ACL Structure**

### 5.99.2.2 Access identifiers

#### 5.99.2.2.1 Access identifiers overview

Initiators are identified in ACEs using one of the following ~~on the basis of one or more of three~~ types of access identifiers ~~(see 7.1)~~:

a) AccessID - based on initiator enrollment as described in 5.99.2.2.2 ~~as enrolled (see 0.0.14) by an initiator using the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 0.0.29)~~;
b) TransportID - based on protocol and interconnect specific identification of initiators as described in 5.99.2.2.3; and
c) vendor specific access identifiers.

#### 5.99.2.2.2 AccessID access identifiers

All AccessID access identifiers shall have the format shown in table 2.

**Table t2 — AccessID access identifier format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ACCESSID | | | | | | | |
| 15 | | | | | | | | |
| 16 | Reserved | | | | | | | |
| 23 | | | | | | | | |

The ACCESSID field contains a value that uniquely identifies the AccessID type ACE in which the AccessID access identifier appears.

~~An AccessID shall be sixteen (16) bytes.~~ An initiator is allowed access to the logical units in an ACE containing an AccessID type access identifier when that initiator is enrolled as described in 5.99.4. An initiator that has not previously enrolled uses the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action to enroll including the AccessID in parameter data as specified in 7.2.4.

An initiator is identified by or associated with an AccessID type access identifier if that initiator is in the enrolled or pending-enrolled state with respect to that AccessID (see 5.99.4). At any given time, an initiator may be identified or associated with at most one AccessID. All initiators enrolled using a given AccessID share the same ACE and access to all the logical units its LUACDs describe.

#### 5.99.2.2.3 TransportID access type identifiers

Use of the TransportID is protocol and interconnect specific.

~~The description of the TransportID and its inclusion in parameter data for parallel SCSI and for SCSI over Fibre Channel initiators is given in 8.99.99.3 and 8.99.99.2, respectively.~~

An initiator is identified by a TransportID if that initiator accessed the SCSI target device ~~(sent any SCSI command)~~ with that TransportID. At any given time, an initiator may be identified or associated with at most one TransportID ~~and by at most one AccessID. Multiple initiators may be associated with the same AccessID~~.

~~Other~~ Protocol standards ~~may~~ should specify the description and use of the TransportID. A protocol specification for a TransportID shall only include ~~address objects~~ initiator port identifiers or names (see SAM-2) that persist across common reset events in the service delivery subsystem. Additionally, a TransportID shall be no more than ~~twenty-four (~~24~~)~~ bytes long and shall have in its first byte a value which uniquely identifies the transport protocol (see 8.99.99).

TransportIDs that for some protocols may be specified in clause 8.99.99 to support protocol standards that are unable to include a TransportID definition. When a protocol standard includes a TransportID definition, the definition in the protocol standard supersedes any definition appearing in this standard.

#### 5.99.2.3 Logical unit access control descriptors

Each LUACD in an ACE identifies one logical unit to which the initator(s) associated with the access identifier are allowed access and specifies the LUN value those initiators use when accessing the logical unit.

The identification of a logical unit~~s~~ in ~~an accessible logical unit pair~~ a LUACD is vendor specific. The LUN value shall conform to the requirements specified in SAM-2.

A logical unit shall be referenced in ~~at most~~ no more than one LUACD per ACE ~~accessible logical unit pair per ACL entry~~. A given LUN value shall appear in ~~at most~~ no more than one LUACD per ACE ~~accessible logical unit pair per ACL entry~~.

### 5.99.3 Managing the ACL

#### 5.99.3.1 ACL management overview

The contents of the ACL are managed by an application client ~~via~~ using the ACCESS CONTROL OUT command with MANAGE ACL and DISABLE ACCESS CONTROLS service actions. The ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2) is used to add, remove, or modify ACEs thus adding, revoking, or changing the allowed access of initiators to logical units. The ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS service action (see 7.2.3) disables access controls and discards the ACL.

#### 5.99.3.2 Authorizing ACL management

To reduce the possibility of applications other than authorized ACL managers changing the ACL, successful completion of ~~these service actions~~ the ACCESS CONTROL OUT command with MANAGE ACL or DISABLE ACCESS CONTROLS service action requires delivery of the correct management identifier key value (see 5.99.7) in the ACCESS CONTROL OUT parameter data. For similar reasons other ACCESS CONTROL OUT and

ACCESS CONTROL IN service actions require the correct management identifier key as summarized in table t3 and table t4.

**Table t3 — ACCESS CONTROL OUT management identifier key requirements**

| Service Action | Name | Management Identifier Key Required | Reference |
|---|---|---|---|
| 00h | MANAGE ACL | Yes | 7.2.2 |
| 01h | DISABLE ACCESS CONTROLS | Yes | 7.2.3 |
| 02h | ACCESS ID ENROLL | No | 7.2.4 |
| 03h | CANCEL ENROLLMENT | No | 7.2.5 |
| 04h | CLEAR ACCESS CONTROLS LOG | Yes | 7.2.6 |
| 05h | MANAGE OVERRIDE LOCKOUT TIMER | Yes/No | 7.2.7 |
| 06h | OVERRIDE MGMT ID KEY | No | 7.2.8 |
| 07h | REVOKE PROXY TOKEN | No | 7.2.9 |
| 08h | REVOKE ALL PROXY TOKENS | No | 7.2.10 |
| 09h | ASSIGN PROXY LUN | No | 7.2.11 |
| 0Ah | RELEASE PROXY LUN | No | 7.2.12 |
| 0Bh-17h | Reserved | | |
| 18h-1Fh | Vendor-specific | | |

**Table t4 — ACCESS CONTROL IN management identifier key requirements**

| Service Action | Name | Management Identifier Key Required | Reference |
|---|---|---|---|
| 00h | REPORT ACL | Yes | 7.1.2 |
| 01h | REPORT LU DESCRIPTORS | Yes | 7.1.3 |
| 02h | REPORT ACCESS CONTROLS LOG | Yes | 7.1.4 |
| 03h | REPORT OVERRIDE LOCKOUT TIMER | Yes | 7.1.5 |
| 04h | REQUEST PROXY TOKEN | No | 7.1.6 |
| 05h-17h | Reserved | | |
| 18h-1Fh | Vendor-specific | | |

~~…shared by the managing application client and the access controls coordinator (see x.x.x, 0.0.27 and also x.x.x). The purpose of the Management Identifier Key is to identify the application client that is responsible for managing access controls for this device.~~

~~NOTE Use of the Management Identifier Key has the following features:~~

~~a) Management of access controls is associated with an application client and not with a particular initiator.~~
~~b) Only an application client that has knowledge of this key may (in most cases) change the ACL for this device; consequently, responsibility for management of access controls may be localized to specific application clients.~~

~~**AUTHOR'S NOTE**: *Is there a better way to rephrase this NOTE?*~~

~~A device has **access controls disabled** when it is shipped from the factory and after successful completion of the ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS. In this state, the ACL is empty (has no entries) and the Management Identifier Key is zero.~~

### 5.99.3.3 Identifying logical units during ACL management

Although the identification of logical units in the ACL is vendor specific (see 5.99.2.3), the ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2) needs a mechanism for identifying logical units that is independent of LUN value and suitable for exchanges between the access controls coordinator and application clients. To serve the needs of the ACCESS CONTROL OUT command with MANAGE ACL service action the access controls coordinator shall identify every logical unit of a SCSI target device ~~shall be identified by~~ with a unique default LUN value. The default LUN values used by the access controls coordinator shall be the LUN values that would be reported by the ~~in~~ REPORTS LUNS command ~~for that logical unit~~ if access controls were disabled. ~~The default LUN value is used in parameter data of access control commands to uniquely identify logical units.~~

An application client discovers the default LUN values using the ACCESS CONTROL IN command with REPORT LU DESCRIPTORS (see 7.1.3) or REPORT ACL (see 7.1.2) service action and subsequently supplies those default LUN values to the access controls coordinator using the ACCESS CONTROL OUT command with MANAGE ACL service action.

The association ~~of~~ between default LUN values and logical units is managed by the access controls coordinator and may change in ways beyond the scope of this standard ~~access controls~~. To track changes in the association between default LUN values and logical units, the access controls coordinator shall maintain the DLgeneration (Default LUNs Generation) value as described in 5.99.3.4.

### 5.99.3.4 Tracking changes in logical unit identification

The access controls coordinator shall maintain the DLgeneration (Default LUNs Generation) ~~a Default LUNs Generation~~ value ~~(see x.x.x and x.x.x) that shall be used to time-stamp~~ to track changes in the association ~~of~~ between default LUN values and logical units.

When access controls are disabled DLgeneration shall be zero. When access controls are first enabled (see 5.99.1) DLgeneration shall be set to one. While access controls are enabled, the access controls coordinator shall increase DLgeneration by one every time ~~This Default LUNs Generation value shall be increased by one each time~~ the association ~~of~~ between default LUN values ~~to~~ and logical units changes for any reason, including but

~~NOTE Changes in the association of default LUNs to logical units that shall cause incrementing the Default LUNs Generation value includes but is~~ not limited to creation of a new logical unit, deletion of an existing logical unit or a change (delete and recreate) of an existing logical unit.

The access controls coordinator shall include the current DLgeneration in the parameter data returned by each ACCESS CONTROL IN command with REPORT LU DESCRIPTORS (see 7.1.3) or REPORT ACL (see 7.1.2) service action. The application client shall supply the DLgeneration for the default LUN values it is using in each ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2). 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 in the parameter data matches the DLgeneration currently in use. If the DLgeneration verification finds a mismatch, the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

~~The first successful ACCESS CONTROL OUT command with MANAGE ACL service action enables access controls (see 0.0.27), that is, transitions the device to the **access controls enabled** state. In this state, all logical units are inaccessible to all initiators unless there are specific rights granted to specific initiators as defined in the ACL.~~

~~An initiator has access to a logical unit as specified in the ACL if that initiator is identified by an access identifier (see x.x.x) in an ACL entry and that logical unit is referenced in an accessible logical unit pair in that ACL entry. In this case, the LUN value for this logical unit as would be returned in parameter data for a REPORT LUNS~~

~~command from this initiator shall be the LUN value of the accessible logical unit pair in that ACL entry. Additionally, an initiator may gain access to a logical unit with a proxy token (see 0.0.19).~~

~~An initiator is identified by or associated with an AccessID identifier if that initiator is in the enrolled or pending-enrolled state with respect to that AccessID (see x.x.x). An initiator is identified by a TransportID if that initiator accessed the device (sent any SCSI command) with that TransportID.~~

## 4.2 Resource requirements for Access Controls

If a device supports the access controls, then the device shall contain an access controls coordinator that shall be able to maintain the following data structures:

a) an ACL consisting of at least one entry where each entry shall contain at least one accessible logical unit pair (see x.x.x);
b) an 8-byte (64 bit) integer called the Management Identifier Key (see x.x.x and 0.0.27);
c) a 4-byte (32 bit) integer called the Default LUNs Generation (see x.x.x);
d) a 2-byte (16 bit) integer called the Initial Override Lockout Timer (see x.x.x);
e) a log of access controls related events containing at least the following (see x.x.x):
   A) a 2-byte (16 bit) integer called the Key Overrides Counter;
   B) a 2-byte (16 bit) integer called the Invalid Keys Counter;
   C) a 2-byte (16 bit) integer called the ACL LUN Conflicts Counter.

Optionally, the access controls coordinator may maintain additional data structures to manage proxy tokens for some or all of the device's logical units (see 0.0.19).

When shipped from the factory, the ACL is empty, all integer values are zero, additional access control log structures are empty and there are no valid proxy tokens.

Persistence of these data structures through power-cycles or target resets is described in x.x.x.

## 4.3 Access Identifiers

Initiators are identified in ACL entries on the basis of one or more of three types of access identifiers (see x.x.x):

a) **AccessID**, as enrolled (see 0.0.14) by an initiator using the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 0.0.29);
b) **TransportID**, protocol and interconnect-specific;
c) vendor-specific identifiers.

An AccessID shall be sixteen (16) bytes. AccessIDs are included in parameter data as specified in 0.0.39.

Use of the TransportID is protocol and interconnect-specific. The description of the TransportID and its inclusion in parameter data for parallel SCSI and for SCSI over Fibre Channel initiators is given in 7.1.3 and 7.1.4, respectively. Other protocol standards may specify the description and use of the TransportID. A protocol specification for a TransportID shall only include address objects that persist across common reset events in the service delivery subsystem. Additionally, a TransportID shall be no more than twenty-four (24) bytes long and shall have in its first byte a value which uniquely identifies the transport protocol (see 0.0.38).

At any given time, an initiator may be identified or associated with at most one TransportID and by at most one AccessID. Multiple initiators may be associated with the same AccessID.

### 5.99.4 Enrolling AccessIDs

### 5.99.4.1 Enrollment states

### 5.99.4.1.1 Summary of enrollment states

Initiators ~~may~~ enroll an AccessID with an access controls coordinator in order to ~~gain~~ be allowed access to logical units ~~accessible via such an~~ listed in the ACE having the same AccessID type access identifier. Enrolling an AccessID is accomplished using the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 7.2.4). An initiator shall be in one of three states with respect to such an enrollment:

a) **not-enrolled**: The state for an initiator before it sends the first ACCESS CONTROL OUT command with ACCESS ID ENROLL service action to the access controls coordinator. ~~when it first accesses the device and~~ Also the state ~~entered into by the~~ for an initiator ~~in response to~~ following successful completion of an ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action (see 7.2.5);
b) **enrolled**: The state for an initiator ~~enters as a consequence of a~~ following successful completion of an ACCESS CONTROL OUT command with ACCESS ID ENROLL service action ~~(see 0.0.29)~~; or
c) **pending-enrolled**: The state for an enrolled initiator following:
   A) ~~enters from the enrolled state as a consequence of certain~~ Events in the service delivery subsystem described in 5.99.11; or
   B) ~~by~~ Successful completion of an ACCESS CONTROL OUT command~~s~~ with MANAGE ACL service action and FLUSH bit set to one (see 7.2.2).

~~The next three subclauses describe these states in more detail and the additional mechanisms that produce transitions between them.~~

### 5.99.4.1.2 Not-enrolled state

The access controls coordinator shall place an initiator ~~enters~~ in the not-enrolled state when it first detects the receipt of a SCSI command or task management function from that initiator ~~accesses the device (sends any SCSI command)~~. ~~An~~ The initiator shall remain in the not-enrolled ~~stays in this~~ state until ~~it~~ successful~~ly completes the~~ completion of an ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 7.2.5).~~ See 0.0.14.3 and 0.0.29.~~

When in the not-enrolled state, an initiator shall only have access to logical units on the basis of a TransportID ~~for that initiator (if that TransportID is an access identifier in an ACL entry)~~ or on the basis of proxy tokens.

The access controls coordinator shall change an initiator ~~in~~ from the enrolled or pending-enrolled state ~~shall transition~~ to the not-enrolled state ~~as follows~~ in response to the following events:

a) ~~by~~ Successful completion of the ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action (see 7.2.5);
b) ~~as a consequence of a~~ Successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2) that ~~replaced an~~ replaces the ACL entry for the enrolled AccessID as follows:
   A) If the NOCNCL bit (see 7.2.2.2) is set to zero in the ACCESS CONTROL OUT command with MANAGE ACL service action parameter data, the state shall change to not-enrolled; or
   B) ~~and, vendor-specifically,~~ If the NOCNCL bit is set to one, the state may change to not-enrolled based on vendor specific criteria ~~(see 6.2.2.2.2)~~; or
c) ~~as a consequence of~~ Power cycles or target resets based on vendor specific criteria (see 5.99.11). ~~in a vendor-specific manner (see x.x.x);~~

~~NOTE 1 An initiator's enrollment transition to the not-enrolled state may be a result of actions not taken by that initiator, but by actions taken by a third-party on behalf of an application client~~ An enrolled initiator may find itself in the not-enrolled state as a result of actions taken by a third-party (e.g., an ACCESS CONTROL OUT command with MANAGE ACL service action performed by another initiator or a target reset). ~~This is in contrast to the CANCEL ENROLLMENT service action which is an action taken by the initiator itself.~~

~~NOTE 2~~ The purpose of ~~this transitioning to the not-enrolled state as a consequence of the MANAGE ACL service action~~ placing an enrolled initiator in the not-enrolled state in response to these events is to give the initiator ~~provide~~ an indication ~~to the initiator~~ that ~~its access rights have~~ the ACE defining its logical unit access has changed.~~, and consequently that its LUN addressing of logical units may have changed, by events or actions not taken by that initiator directly.~~ One consequence of changes in an ACE is that previous relationships between logical units and LUN values may no longer apply. ~~The use of the MANAGE ACL service action by the managing application client should be coordinated with the affected initiators to ensure proper data integrity. Such coordination is beyond the scope of this standard.~~

~~An enrolled initiator may find itself in not-enrolled state as a result of actions taken by a third-party~~

If an initiator detects this loss of enrollment, it may ~~then~~ take ~~the appropriate~~ recovery actions. However, such actions may be disruptive for the initiator and may not always be required. Use of the not-enrolled state and the resulting disruptive recovery actions are avoidable if the application client that sends the ACCESS CONTROL OUT command with MANAGE ACL service action is able to determine its requested changes to the ACL do not alter the existing relationships between logical units and LUN values in any existing ACEs with AccessID type access identifiers.

If the application client that sends the ACCESS CONTROL OUT command with MANAGE ACL service action is unable to determine whether the ACE logical unit relationships are altered as a result of processing the command, then it should set the NOCNCL bit to zero and it should coordinate the ACL change ~~The use of the MANAGE ACL service action by the managing application client should be coordinated~~ with the affected initiators to ensure proper data integrity. Such coordination is beyond the scope of this standard.

~~If the managing application client determines that these recovery actions are not required, the application client~~ If the application client that sends the ACCESS CONTROL OUT command with MANAGE ACL service action is able to determine that ACE logical unit relationships are not be altered as a result of processing the command, then it should set the NOCNCL bit to one, recommending ~~to recommend~~ to the access controls coordinator that ~~it leave the~~ initiators be left in their ~~in its~~ current enrollment states.

~~A vendor~~ The access controls coordinator has at least three ~~implementation~~ vendor specific options for responding to a NOCNCL bit value of one ~~the access controls coordinator~~:

a) Honor the recommendation. ~~(~~This is least disruptive for the initiator and requires no extra actions on the part of the access controls coordinator~~).~~;
b) Ignore the recommendation and always ~~transition~~ place the initiator in the non-enrolled state. ~~(~~This may disrupt an initiator unnecessarily, but requires no extra resources on the part of the access controls coordinator~~).~~; or
c) Ignore the recommendation and instead examine the current and new ACEs ~~access rights and LUN addressing~~ to ~~(independent of the managing application client)~~ determine if the initiator should be ~~transitioned~~ placed in the non-enrolled state. ~~In other words, independently take the responsibility from the managing application client.~~

~~These recovery actions on the part of the initiator are typically not required if, for all accessible logical units for which access rights are left unchanged, the LUN addressing also does not change. That is, LUNs may be added or deleted from the initiator's REPORT LUNS parameter list, but any value in both the list prior to the change and after the change still addresses the same logical unit.~~

~~When in the not-enrolled state, an initiator shall only have access to logical units on the basis of a TransportID for that initiator (if that TransportID is an access identifier in an ACL entry) or on the basis of proxy tokens.~~

### 5.99.4.1.3 Enrolled state

The access controls coordinator shall place an initiator ~~enters~~ in the enrolled state (or enrolls the initiator) ~~from either the not-enrolled or pending-enrolled state by~~ following successful completion of the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 7.2.4). ~~This~~ The ACCESS CONTROL OUT command with ACCESS ID ENROLL service action is successful only under the following conditions:

a) If the initiator was in the not-enrolled state and the AccessID in the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action parameter data ~~of the service action~~ matches the access identifier in an ACE. ~~entry in the ACL (so that this AccessID has rights to one or more logical units); in this case,~~ The initiator thus enrolled is allowed ~~gains~~ access to ~~those~~ the logical units specified in the LUACDs in the ACE (see 5.99.2) ~~access grant logical unit pairs of that ACL entry~~; or
b) If the initiator was in the enrolled or pending-enrolled state and the AccessID in the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action parameter data matches ~~that of~~ the current enrolled AccessID for ~~that~~ the initiator~~; in this case, commands to the affected logical units are handled according to the rules of 4.10~~.

If the initiator was in the enrolled or pending-enrolled state and the AccessID in the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action parameter data does not match ~~that of~~ the current enrolled AccessID for ~~that~~ the initiator, the ~~device server shall respond~~ command shall be terminated with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense code shall be set to ACCESS DENIED - ENROLLMENT CONFLICT, and ~~as specified in 0.0.29. Additionally,~~ the access controls coordinator shall transition an enrolled initiator to the pending-enrolled state.

~~The AccessID enrollment of an initiator (in either the enrolled or not-enrolled state) may be kept in non-volatile memory in a vendor-specific manner subject to the rules in x.x.x.~~

Transitions out of the enrolled state are described in:~~ the subclauses for the not-enrolled (0.0.14.2) and pending-enrolled (0.0.14.4)states.~~

a) 5.99.4.1.2 for changes to the not-enrolled state; and
b) 5.99.4.1.4 for changes to the pending-enrolled state.

NOTE 1 - This standard does not preclude implicit enrollments through mechanisms in the service delivery subsystem. Such mechanisms should perform implicit enrollments after identification by TransportID and should fail in the case where there are ACL conflicts as described in 5.99.4.2.

### 5.99.4.1.4 Pending-enrolled state

The access controls coordinator shall place an initiator ~~shall enter~~ in the pending-enrolled state only ~~from~~ if that initiator currently is in the enrolled state, and ~~as a consequence of~~ in response to the following:

a) Any event in the service delivery subsystem that causes the access controls coordinator to question whether an initiator in the enrolled state has changed its AccessID (e.g., a ~~PRLO or LOGO~~ process or port logout in ~~FCP~~ Fibre Channel, or a hard bus reset for parallel SCSI);
b) Successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action ~~and~~ where the FLUSH bit is set to one in the parameter data; or
c) Optionally after a TARGET RESET task management function, as described in 5.99.11.

While in the pending-enrolled state, the initiator’s access to logical units is limited ~~according to the rules of~~ as described in 5.99.6.

#### 5.99.4.2 ACL LUN conflict resolution

Three types of ACL LUN conflicts may occur at the time an initiator in the not-enrolled state attempts to enroll an AccessID ~~by~~ using the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action:

a) The TransportID ~~entry~~ ACE (see 5.99.2) and the AccessID ~~entry~~ ACE for the initiator ~~in the ACL~~ each contain a LUACD ~~an accessible logical unit pair~~ with the same LUN value but with references to different logical units;
b) The TransportID ~~entry~~ ACE and the AccessID ~~entry~~ ACE for the initiator ~~in the ACL~~ each contain a LUACD ~~an accessible logical unit pair~~ with the different LUN values but with references to the same logical unit; or
c) The initiator has proxy access rights to a logical unit addressed with a LUN value that equals a LUN value in a LUACD in ~~an accessible logical unit pair of~~ the AccessID ~~entry~~ ACE for the initiator ~~in the ACL~~.

In any of these cases, the following actions shall be taken as part of the handling of the enrollment function ~~service action~~:

a) The ACCESS CONTROL OUT command with ACCESS ID ENROLL service action shall be terminated with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense code shall be set to ACCESS DENIED - ACL LUN CONFLICT;
b) The ~~access controls coordinator shall fail the enrollment and leave the~~ initiator shall remain in the not-enrolled state; and
c) Except when the ACL LUN conflict is the result of proxy access rights ~~in the last case~~, the access controls coordinator shall record the event in the access controls log as described in 5.99.9.
~~d) the device server shall return status and sense data indicating that a conflict arose and that the enrollment failed (see 0.0.29).~~

### 5.99.5 Granting and revoking access rights

#### 5.99.5.1 Non-proxy access rights

The ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2) adds or replaces ACEs in the ACL ~~entries~~ (see 5.99.2). One ACE ~~ACL entry~~ describes the logical unit access allowed to one access identifier (see 5.99.2.2) and the LUN values to be used in addressing the accessible logical units. The access identifier designates the ~~for~~ initiator(s) that may be permitted the logical unit access described by the ACE.~~ associated with the access identifier.~~

With the exception of proxy access rights (see 5.99.5.2), logical unit access rights are granted by:

~~a)~~ Adding a new ACE to the ACL allowing logical unit access to a new access identifier ~~new ACL entry for an access identifier~~; or ~~by~~
b) Replacing an existing ACE ~~ACL entry for an access identifier~~ so that the revised ACE ~~ACL entry~~ includes additional LUACDs ~~accessible logical unit pairs~~.

With the exception of proxy access rights, access rights are revoked by:

a) Removing an ACE from the ACL ~~entry for an access identifier or by~~; or
b) Replacing an existing ACE ~~ACL entry for an access identifier~~ so that the revised ACE ~~ACL entry excludes~~ removes one or more LUACDs ~~prior accessible logical unit pairs~~.

~~If an ACL entry is added or replaced the rules of x.x.x and 0.0.14.2 shall apply.~~ When an ACE is added or replaced the requirements stated in 5.99.4.1.2 and 5.99.10 apply.

### 5.99.5.2 Proxy ~~tokens and proxy~~ access

#### 5.99.5.2.1 Proxy tokens

An initiator with access to a logical unit on the basis of ~~either a TransportID or AccessID~~ an ACE in the ACL (see 5.99.5.1) may temporarily share that access with third parties ~~via~~ using the proxy mechanism. The initiator uses the ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action (see 7.1.6) to request~~s~~ that the access control coordinator generate a proxy token for the logical unit specified by the LUN value in the parameter data ~~from the access controls coordinator a Proxy Token for a specific logical unit via the ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action (see 0.0.25)~~.

The access controls coordinator generates ~~this~~ the proxy token in ~~an implementation-specific~~ a vendor specific manner. ~~NOTE~~ For a given SCSI target device, all active proxy token values should be unique. ~~Also,~~ Proxy token values should not be reused any more frequently than is necessary to prevent stale proxy tokens from being given unintended meaning.

~~As long as the Proxy Token remains valid and no power-cycles or target resets have occurred, this proxy access right is unchanged.~~

~~A Proxy Token shall be made invalid by the following events:~~

Power cycles and target resets shall not affect the validity and proxy access rights of proxy tokens (see 5.99.11). A proxy token shall remain valid and retain the same proxy access rights until one of the following occurs:

a) An initiator with access to the logical unit based on an ACE in the ACL revokes the proxy token using:
   A) ~~by~~ The ACCESS CONTROL OUT command with ~~the~~ REVOKE PROXY TOKEN service action (see 7.2.9) supplying ~~naming~~ the specific proxy token in the parameter data; or
   B) ~~with~~ The ACCESS CONTROL OUT command with REVOKE ALL PROXY TOKENS service action (see 7.2.10) ~~(see 0.0.34 and 0.0.35)~~;
b) An application client issues the ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2) with parameter data containing the ~~and appropriate~~ Revoke Proxy Token page (see 7.2.2.4) or Revoke All Proxy Tokens ~~parameter~~ page~~s~~ (see 7.2.2.5).

#### 5.99.5.2.2 Proxy LUNs

The initiator ~~then~~ forwards the proxy token (see 5.99.5.2.2) to a third party (e.g., in a target descriptor in the parameter data of the EXTENDED COPY command~~; see Appendix C.2~~).

The third party ~~then~~ sends the access controls coordinator an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 7.2.11) containing the proxy token to request creation of a proxy access right to the referenced logical unit ~~at the requested LUN (see 0.0.36)~~. The access controls coordinator determines the referenced logical unit from the proxy token value; the third party is unaware of the exact logical unit to which it is requesting access. The parameter data for the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action includes the LUN value that the third party intends to use when accessing the referenced logical unit. The LUN value thus assigned is called a proxy LUN.

~~As long as the Proxy Token remains valid and no power-cycles or target resets have occurred, this proxy access right is unchanged.~~

~~A Proxy Token shall be made invalid by the following events:~~

~~a) an initiator with access to the logical unit revokes the Proxy Token by the ACCESS CONTROL OUT command with the REVOKE PROXY TOKEN service action naming the specific Proxy Token or with the REVOKE ALL PROXY TOKENS service action (see 0.0.34 and 0.0.35);~~
~~b) an application client issues the ACCESS CONTROL OUT command with MANAGE ACL service action and appropriate Revoke Proxy Token or Revoke All Proxy Tokens parameter pages (see 0.0.27.2.3).~~

A proxy LUN ~~(i.e., a LUN associated to a logical unit on the basis of a proxy access right resulting from a successful ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action)~~ shall ~~be~~ remain valid until ~~unless~~ one of the following occurs:

a) The third party releases the proxy LUN value ~~with~~ using the ACCESS CONTROL OUT command ~~and~~ with RELEASE PROXY LUN service action (see 7.2.12);
b) An event in the service delivery subsystem causes the access controls coordinator to question whether the third party initiator that created the proxy LUN value has changed ~~(~~and may no longer be in possession of the proxy token);
c) The proxy token is made invalid as described in 5.99.5.2.1; or
d) A power cycle or target reset occurs (see 5.99.11).

~~In the latter two cases,~~ If the third party believes that the invalidation of a proxy LUN value is temporary, it may reissue the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action in an attempt to re-establish its proxy access rights. ~~In the last case, the access controls coordinator shall fail the request to re-establish proxy access rights.~~ The access controls coordinator shall process the request as described in 5.99.5.2.1 without reference to any previous assignment of the proxy LUN value.

### 5.99.6 Verifying access rights ~~for initiators~~

When ~~the access controls coordinator has~~ access controls are enabled (see 5.99.1), access rights ~~from a given~~ for an initiator ~~are~~ shall be validated ~~in the following manner~~ as described in this subclause.

All commands ~~to a specific logical unit via a specific LUN value are~~ shall be processed as if access controls were not present if the ACL (see 5.99.2) allows the initiator ~~has~~ access to the addressed logical unit by virtue of one of the following conditions:

a) ~~An~~The ACL contains an ACE containing a TransportID type access identifier ~~ACL entry~~ for ~~that~~ the initiator and that ACE includes a LUACD ~~an accessible logical unit pair~~ with LUN value matching the addressed LUN;
b) The initiator is in the enrolled state (see 5.99.4.1.3) under an AccessID, ~~and~~ the ACL contains an ACE containing that AccessID as an access identifier, ~~has an ACL entry~~ and that ACE includes a LUACD ~~an accessible logical unit pair~~ with LUN value matching the addressed LUN; or
c) The addressed LUN matches a proxy LUN value (see 5.99.5.2.2) assigned ~~via a valid proxy token via~~ using the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 7.2.11) and the proxy token (see 5.99.5.2.1) used to assign the proxy LUN value is still valid.

If the initiator is in the pending-enrolled state (see 5.99.4.1.4) under an AccessID, the ACL contains an ACE containing that AccessID as an access identifier, and that ACE includes a LUACD with LUN value matching the addressed LUN ~~has access to the logical unit by virtue of an AccessID enrolled by that initiator and the initiator is in the pending-enrolled state~~, then commands shall be processed as follows:

a) INQUIRY, REPORT LUNS, ACCESS CONTROL OUT and ACCESS CONTROL IN commands shall be processed as if access controls were not present;
b) All other commands shall be terminated ~~prior to any data transfer~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INITIATOR PENDING-ENROLLED.

~~NOTE~~ An initiator should respond to the ACCESS DENIED - INITIATOR PENDING-ENROLLED additional sense code by sending an ACCESS CONTROL OUT command with ACCESS ID ENROLL service action. If the command succeeds, the initiator may retry the failed command.

If an INQUIRY command is addressed to a LUN for which there is no matching LUN value in any LUACD in any ACE allowing the initiator logical unit access rights, ~~that is not associated for that initiator to an accessible logical unit, the device server shall set~~ the standard INQUIRY data (see 7.z.z) PERIPHERAL DEVICE TYPE field shall be set to 1Fh and the PERIPHERAL QUALIFIER field shall be set to 011b (the device server is not capable of supporting a device at this logical unit).

The parameter data returned in response to a REPORT LUNS command addressed to LUN 0 shall return only the list of LUN values that are associated to accessible logical units according to the following criteria:

a) If the initiator is in the enrolled or pending-enrolled state, ~~this list~~ the REPORT LUNS parameter data shall include any LUN values ~~associated to accessible logical units by virtue of~~ found in LUACDs in the ACE containing the AccessID enrolled by ~~that~~ the initiator;
b) If the initiator (in any enrollment state) has a TransportID found in the access identifier of an ACE, the REPORT LUNS parameter data shall include any LUN values found in LUACDs in that ACE; and
c) If the initiator (in any enrollment state) has access to any proxy LUNs (see 5.99.5.2.2) ~~logical units by virtue of proxy tokens~~, ~~the corresponding~~ those LUN values ~~are also~~ shall be included in the REPORT LUNS parameter data.

If the initiator is ~~(~~in the not-enrolled state~~) has no access rights~~ and is not allowed access to any logical unit ~~(either through a~~ as result of its TransportID or as a result of a proxy LUN assignment ~~through a Proxy Token)~~, then the ~~response to~~ REPORT LUNS parameter data shall include only LUN 0, as specified in 7.z.z~~SPC-3 rev 0, 7.19~~.

~~**AUTHOR'S NOTE:** *The reference above will need to be checked after this clause is inserted into SPC-3.*~~

Except when access controls are disabled, all cases not described previously in this subclause shall result in termination of the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to LOGICAL UNIT NOT SUPPORTED.

## 4.6 Granting and revoking access rights

### 4.6.1 Non-proxy access rights

The ACCESS CONTROL OUT command with MANAGE ACL service action adds or replaces ACL entries (see x.x.x and 0.0.27). One ACL entry describes the access allowed to one access identifier (see x.x.x), and the LUN values to be used in addressing the accessible logical units for initiators associated with the access identifier.

With the exception of proxy access rights (see 0.0.19), access rights are granted by adding a new ACL entry for an access identifier or by replacing an existing ACL entry for an access identifier so that the revised ACL entry includes additional accessible logical unit pairs. Access rights are revoked by removing an ACL entry for an access identifier or by replacing an existing ACL entry for an access identifier so that the revised ACL entry excludes one or more prior accessible logical unit pairs.

If an ACL entry is added or replaced the rules of x.x.x and 0.0.14.2 shall apply.

### 4.6.2 Proxy tokens and proxy access

An initiator with access to a logical unit on the basis of either a TransportID or AccessID may temporarily share that access with third parties via the proxy mechanism.

The initiator requests from the access controls coordinator a Proxy Token for a specific logical unit via the ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action (see 0.0.25). The access controls coordinator generates this Proxy Token in an implementation-specific manner.

NOTE All active Proxy Token values should be unique. Also, Proxy Token values should not be reused any more frequently than is necessary to prevent stale Proxy Tokens from being given unintended meaning.

The initiator then forwards the Proxy Token to a third party (e.g., in a target descriptor in the parameter data of the EXTENDED COPY command; see Appendix C.2).

The third party then sends the access controls coordinator an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action containing the Proxy Token to request creation of a proxy access right to the referenced logical unit at the requested LUN (see 0.0.36).

As long as the Proxy Token remains valid and no power-cycles or target resets have occurred, this proxy access right is unchanged.

A Proxy Token shall be made invalid by the following events:

a) an initiator with access to the logical unit revokes the Proxy Token by the ACCESS CONTROL OUT command with the REVOKE PROXY TOKEN service action naming the specific Proxy Token or with the REVOKE ALL PROXY TOKENS service action (see 0.0.34 and 0.0.35);
b) an application client issues the ACCESS CONTROL OUT command with MANAGE ACL service action and appropriate Revoke Proxy Token or Revoke All Proxy Tokens parameter pages (see 0.0.27.2.3).

A proxy LUN (i.e., a LUN associated to a logical unit on the basis of a proxy access right resulting from a successful ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action) shall be valid unless one of the following occurs:

a) the third party releases the LUN value with the ACCESS CONTROL OUT command and RELEASE PROXY LUN service action (see 0.0.37);
b) an event in the service delivery subsystem causes the access controls coordinator to question whether the third party initiator that created the LUN value has changed (and may no longer be in possession of the Proxy Token).
c) the Proxy Token is made invalid, as above;

In the latter two cases, the third party may reissue the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action in an attempt to re-establish its proxy access rights. In the last case, the access controls coordinator shall fail the request to re-establish proxy access rights.

## 4.5 Interactions of Access Controls and other features

### 4.5.1 Queuing Relationships and Access Controls

Upon successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action, the new access control state defined by that command shall apply to all tasks that subsequently enter the task enabled state. Tasks that have modified the media, mode pages, or equivalent target elements shall not be affected by an

ACCESS CONTROL OUT command that subsequently enters the task enabled state. Tasks in the task enabled state that have not modified the media, mode pages or equivalent target elements may or may not be affected by an ACCESS CONTROL OUT command that subsequently enters the task enabled state. The access control state in effect prior to when the ACCESS CONTROL OUT command (with MANAGE ACL or DISABLE ACCESS CONTROLS service action) entered the task enabled state shall apply to all tasks that are not affected by the ACCESS CONTROL OUT command.

NOTE A task completes all its media modifications etc. under the control of a single access control state, either the state in effect prior to processing of the ACCESS CONTROL OUT command or the state in effect following processing of the ACCESS CONTROL OUT command. Once a task has begun its media modifications etc., changes in the access control state have no effect on the task.

Multiple access control commands (both ACCESS CONTROL IN and ACCESS CONTROL OUT) may be queued at the same time. The order of processing of such commands is defined by the tagged queuing restrictions, if any, but each is processed as a single indivisible command without any interleaving of actions that may be required by other access control commands.

### 4.5.2 Existing reservations and ACL changes

If a logical unit is reserved by one initiator and that logical unit becomes accessible to another initiator as a consequence of any access control command, there shall be no changes in the reservation state of that logical unit.

If a logical unit is reserved by an initiator and that logical unit becomes inaccessible to that initiator as a consequence of any access control command or other event, there shall be no changes in the reservation. Existing mechanisms in RESERVE/RELEASE and Persistent Reservations allow for other initiators with access to that logical unit to clear the reservation.

### 5.99.7 The management identifier key

#### 5.99.7.1 Management identifier key usage

The purpose of the management identifier key is to identify the application ~~client~~ that is responsible for managing access controls for ~~this~~ a SCSI target device. This identification is accomplished by allowing the application client to specify a new management identifier key value in the parameter data of each ACCESS CONTROL OUT command with the MANAGE ACL service action (see 7.2.2), and by requiring the most recently specified management identifier key value to appear in many ACCESS CONTROL IN and ACCESS CONTROL OUT service actions.

To allow for failure scenarios where the management identifier key value has been lost, an override procedure involving a timer is provided as described in 5.99.7.2.

~~…shared by the managing application client and the access controls coordinator (see x.x.x, 0.0.27 and also x.x.x).~~

~~NOTE~~ Use of the management identifier key has the following features:

a) Management of access controls is associated with ~~an~~ those application clients that are able to provide the correct management identifier key and not with a ~~particular~~ single initiator port identifier (see SAM-2);
b) Only an application client that has knowledge of ~~this~~ the management identifier key may (in most cases) change the ACL for ~~this~~ the SCSI target device~~; consequently, responsibility for~~ with the result that management of access controls may be ~~localized~~ limited to specific applications and application clients.

~~**AUTHOR'S NOTE**: *Is there a better way to rephrase this NOTE?*~~

### 5.99.7.2 Overriding ~~of~~ the management identifier key

#### 5.99.7.2.1 The OVERRIDE MGMT ID KEY service action

~~The management identifier key is required for successful processing of many of the ACCESS CONTROL IN and ACCESS CONTROL OUT command service actions (e.g., REPORT ACL and MANAGE ACL). Each ACCESS CONTROL OUT command with MANAGE ACL service action updates the Management Identifier Key. See Table 6 and Table 26 for a summary of the service actions requiring the Management Identifier Key.~~

~~However, C~~Conditions may arise when ~~this key~~ the management identifier key needs to be replaced and the current key is not available. ~~In this case~~ When this occurs, the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action (see 7.2.8) may be used to force **the** management identifier key to a known value.

The ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action is intended only for failure scenarios. The ACCESS CONTROL OUT command with MANAGE ACL service action should be used in all other circumstances.

To ~~facilitate~~ protect~~ion of~~ the management identifier key from unauthorized overrides, the access controls coordinator shall ~~support the following.~~ restrict use of the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action based on the value of the override lockout timer (see 5.99.7.2.2).

When ~~this~~ the override lockout timer is ~~non-~~ not zero, an ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action shall ~~fail~~ be terminated with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

~~If this timer is zero, then the~~ When the override lockout timer is zero, an ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action shall ~~succeed~~ be processed as described in 7.2.8.

~~Both of these events are logged as described in 4.11.~~ The access controls coordinator shall log the receipt of all ACCESS CONTROL OUT commands with OVERRIDE MGMT ID KEY service action and their success or failure as described in 5.99.9.

#### 5.99.7.2.2 The override lockout timer

The access controls coordinator shall maintain the override lockout timer as a 16 bit unsigned integer ~~non-negative integer-valued timer, called the Override Lockout Timer~~. ~~This timer, if non-zero,~~ When the override lockout timer is not zero it shall be decreased by one approximately once per second but no more frequently than once every 800 milliseconds until the value reaches zero. ~~When this timer is non-zero, an OVERRIDE MGMT ID KEY service action shall fail with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB. If this timer is zero, then the OVERRIDE MGMT ID KEY shall succeed. Both of these events are logged as described in 4.11.~~

The ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action manages the state of the override lockout timer (see 7.2.7),~~. This service action has~~ performing one of two functions~~,~~ depending on whether ~~or not~~ the correct management identifier key is supplied in the parameter data.

a) If the incorrect management identifier key is supplied ~~in the parameter data is incorrect (~~or if no parameter data is sent~~)~~, the access controls coordinator shall ~~restart~~ reset the override lockout timer~~, that is, reset it~~ to ~~its current initial~~ the most recently received initial override lockout timer value; or

b) If the correct management identifier key is supplied ~~in the parameter data is correct~~, then the access controls coordinator shall do the following:
   1) ~~reset~~ Save the initial override lockout timer value ~~according to data~~ supplied in the parameter data; and
   2) ~~restart~~ Reset the override lockout timer to the new initial value.

NOTE 2 - Setting the initial override lockout timer value to zero disables the override lockout timer and allows~~ for~~ the ACCESS CONTROL OUT command with OVERRIDE MGMT KEY service action to succeed at any time.

Overloading the management key identifier to have a function selection usage is an unusual operational specification, however, it offers significant advantages for the ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action. Any application that knows the management identifier key may establish an initial override lockout timer value of sufficient duration (up to about 23 hours). Maintaining a non-zero override lockout timer value may be accomplished without knowing the management identifier key or transporting the management identifier key on the service delivery subsystem. Attempts to establish a zero initial override lockout timer value that are not accompanied by the correct management identifier key result in decreasing the probability that a subsequent ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action is able to succeed by resetting the override lockout timer to the most recently specified initial value that was accompanied by the correct management identifier key.

~~This model has the following features:~~

~~a) an application client could easily maintain a positive value for the Override Lockout Timer, since any initiator has the ability to force a restart (no Management Identifier Key is required);~~
~~b) the managing application client (the one that manages the Management Identifier Key) has the ability establish the policy for protecting the key from inadvertent override in a manner consistent with deployment policies;~~
~~c) by reporting the initial and current value, the managing application client may approximately measure the real-time accuracy of the timer used by the access controls coordinator;~~
~~d) by logging all override events, the managing client application may be able to ascertain if an inadvertent override was attempted or occurred and which initiator was involved.~~

~~Additionally,~~ The ACCESS CONTROL IN command with REPORT OVERRIDE LOCKOUT TIMER may be used ~~by the application client~~ to ~~report on~~ discover the state of the override lockout timer.

~~NOTE Setting the Initial Override Lockout Timer value to zero disables the timer and allows for the OVERRIDE MGMT KEY service action to succeed at any time.~~

### 5.99.8 Reporting access control information

Specific service actions of the ACCESS CONTROL IN command may be used by an application client to request a report from the access controls coordinator about its access controls data and state.

The ACCESS CONTROL IN command with REPORT ACL service action (see 7.1.2) returns the ACL (see 5.99.2). The information reported includes the following:

a) the list of access identifiers (see 5.99.2.2) and ~~their access rights~~ the associated LUACDs (see 5.99.2.3) currently in effect; and
b) the list of ~~proxies~~ proxy tokens (see 5.99.5.2.1) currently in effect.

The ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action (see 7.1.4) returns the contents of the access controls log (see 5.99.9).

The ACCESS CONTROL IN command with REPORT OVERRIDE LOCKOUT TIMER service action~~s~~ (see 7.1.5) reports on the state of the override lockout timer (see 5.99.7.2.2).

### 5.99.9 Access controls log

The access controls log is a record of events ~~related to the access controls state~~ maintained by the access controls coordinator.

The access controls log has three portions, recording different classes of events:

a) invalid key events: ~~(when~~ a mismatch between the management identifier key (see 5.99.7) in a CDB or parameter data ~~does not to match~~ and the current value maintained by the access controls coordinator~~)~~;
b) key override events: ~~(when an attempt is made to~~ attempts to override the management identifier key (see 5.99.7.2.1), whether the attempt fails or succeeds~~)~~;
c) ACL LUN conflict events (see 5.99.4.2).

Each portion of the log is required to contain a counter of the events. When a device ships from the factory, the counters shall be zero. The counters ~~are~~ shall be increased by one whenever the relevant event occurs

Optionally, each log portion may contain additional records with more specific information about each event~~ as described in the following paragraphs of this clause~~. ~~If the additional event records resources are exhausted, new records are always prepended to the log and the oldest records are deleted.~~ When the resources for additional log records are exhausted, the access controls coordinator shall preserve the most recently added log records in preference to the older log records.

Log records contain a ~~The content of the~~ TIME STAMP field~~s~~ whose contents ~~described in the log records~~ are vendor specific. If the ~~device~~ access controls coordinator has no time stamp resources the TIME STAMP field~~s~~ 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.

~~The~~ Invalid key events occur whenever an access controls command requires checking ~~a field~~ an initiator supplied management identifier key either in the CDB or in the parameter data against the current management identifier key saved by the access controls coordinator and the two values fail to match ~~this check fails because the value in the field does not equal the current value maintained by the access controls coordinator~~. When such an event occurs, the access controls coordinator shall increase the invalid keys 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 7.1.4.2.3) describing the event. ~~prepend to this portion of the log a record that includes the TransportID of the initiator that sent the command, the operation code of the command and its service action, the invalid key and a 32 bit integer time-stamp. (See 0.0.23.3.)~~

~~The override key~~ Key override events occur when the access controls coordinator receives the ACCESS CONTROL OUT command with OVERRIDE MGMT KEY service action (see 7.2.8). When such an event occurs, the access controls coordinator shall increase the key overrides 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 an key overrides log record (containing the information defined in 7.1.4.2.2) describing the event. ~~prepend to this portion of the log a record that includes the TransportID of the initiator that sent the command, a flag that indicates if the override was successful, the current value and initial setting of the Override Lockout Timer, and a 32 bit integer time-stamp. (See 0.0.23.2.)~~

~~The~~ ACL LUN conflict events occur as specified in 5.99.4.2. When such an event occurs, the access controls coordinator shall increase the ACL LUN conflicts 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 7.1.4.2.4) describing the event. ~~prepend to this portion of the log a record that includes the TransportID of the initiator enrolling the AccessID that created the conflict, and a 32 bit integer time-stamp. (See 0.0.23.4.)~~

~~The content of the time stamp fields described in the log records are vendor specific. If the device has no time stamp resources the fields 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.~~

~~If the additional event records resources are exhausted, new records are always prepended to the log and the oldest records are deleted.~~

Selected portions of the access controls log may be ~~reported to~~ requested by an application client ~~by~~ using the ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action (see 7.1.4). With the exception of the key overrides portion, selected portions of the log may be cleared and the counters reset to zero ~~with~~ using the ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action (see 7.2.6).

### 5.99.10 Interactions of access controls and other features

#### 5.99.10.1 Queuing relationships and access controls

Upon successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action (see 7.2.2), the ~~new access control state~~ ACL (see 5.99.2) defined by that command shall apply to all tasks that subsequently enter the task enabled state. Tasks that have modified the media, mode pages, or equivalent SCSI target device elements shall not be affected by an ACCESS CONTROL OUT command that subsequently enters the task enabled state. Tasks in the task enabled state that have not modified the media, mode pages or equivalent SCSI target device elements may or may not be affected by an ACCESS CONTROL OUT command that subsequently enters the task enabled state. The ~~access control state~~ ACL in effect prior to when the ACCESS CONTROL OUT command ~~(~~with MANAGE ACL or DISABLE ACCESS CONTROLS service action~~)~~ entered the task enabled state shall apply to all tasks that are not affected by the ACCESS CONTROL OUT command.

~~NOTE~~ A task ~~completes~~ shall complete all its media modifications etc. under the control of a single ACL ~~access control state~~, either the state in effect prior to processing of the ACCESS CONTROL OUT command or the state in effect following processing of the ACCESS CONTROL OUT command. ~~Once~~ After a task has begun its media modifications etc., ~~changes in the access control state~~ changing the access control state from disabled to enabled (see 5.99.1) shall have no effect on the task.

Multiple access control commands, both ACCESS CONTROL IN and ACCESS CONTROL OUT, may be queued concurrently ~~at the same time~~. The order of processing of such commands is defined by the tagged queuing restrictions, if any, but each command shall be ~~is~~ processed as a single indivisible command without any interleaving of actions that may be required by other access control commands.

#### 5.99.10.2 Existing reservations and ACL changes

If a logical unit is reserved by one initiator and that logical unit becomes accessible to another initiator as a ~~consequence~~ result of ~~any~~ an access control command, there shall be no changes in the reservation state of that logical unit.

If a logical unit is reserved by an initiator and that logical unit becomes inaccessible to that initiator as a ~~consequence~~ result of ~~any~~ an access control command or other access control related event, there shall be no changes in the reservation. Existing mechanisms in RESERVE/RELEASE and Persistent Reservations allow for other initiators with access to that logical unit to clear the reservation.

### ~~4.8 Preserving access control information (power cycles and target resets)~~

### 5.99.11 Access controls information persistence and memory usage requirements ~~(power cycles and target resets)~~

If a SCSI target device supports the access controls, then the SCSI target device shall contain an access controls coordinator that ~~The access controls coordinator is required to~~ shall maintain the following information in nonvolatile memory: ~~form the entire~~

a) Whether access controls are enabled or disabled;
b) The access controls data ~~as~~ described as persistent across power cycles and resets in table t5 and table t6~~, including the access controls log (see x.x.x)~~.

If access controls are disabled the readiness of the access control coordinator's nonvolatile memory shall not affect the processing of commands. If access controls are enabled and the access control coordinator's nonvolatile ~~If the device's non-volatile~~ memory is not ready ~~(to read the access controls data)~~, the device servers for all logical units shall terminate all commands except INQUIRY commands with ~~return on all addressed logical units~~ a CHECK CONDITION status, ~~a~~ the sense key shall be set to NOT READY and additional sense data shall be set as ~~defined in the TEST UNIT READY command (see SPC-3, rev 0, 7.25) for all commands except INQUIRY~~ described in table 117 (see 7.z.z).

~~Additionally, all valid Proxy Tokens created as a consequence of ACCESS CONTROL IN commands with REQUEST PROXY TOKEN service action (see 0.0.25) shall be preserved through a power-cycle or target reset. However, any proxy access rights created by an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 0.0.36) shall not be preserved.~~

~~It is vendor-specific what effects either a power cycle or target reset may have on initiator enrollment states:~~

~~a) If the access controls coordinator preserves enrollments, then after the reset is complete all initiators formerly in the enrolled or pending-enrolled state enter the pending-enrolled state until changed by an ACCESS CONTROL OUT command with ACCESS ID ENROLL or CANCEL ENROLLMENT service action.~~

~~b) If the access controls coordinator does not preserve enrollments, then after the reset is complete all initiators shall enter the not-enrolled state until changed by an ACCESS CONTROL OUT command with ACCESS ID ENROLL service action.~~

Following a power cycle or reset event, all previously enrolled initiators shall be placed in the same enrollment state and that state shall be one of the following:

a) pending-enrolled (see 5.99.4.1.4); or
b) not-enrolled (see 5.99.4.1.2).

### 4.9 Reporting access control information

Specific service actions of the ACCESS CONTROL IN command may be used by an application client to request a report from the access controls coordinator about its access controls data and state.

The REPORT ACL service action returns the ACL (see x.x.x and x.x.x). The information reported includes the following:

a) the list of access identifiers and their access rights currently in effect;
b) the list of proxies currently in effect.

The REPORT ACCESS CONTROLS LOG service action returns the contents of the access controls log (see x.x.x).

The REPORT OVERRIDE LOCKOUT TIMER service actions reports on the state of the Override Lockout Timer (see x.x.x).

### 4.10 Verifying access rights for initiators

When the access controls coordinator has access controls enabled, access rights from a given initiator are validated in the following manner.

All commands to a specific logical unit via a specific LUN value are processed as if access controls were not present if the initiator has access to the logical unit by virtue of one of the following conditions:

a) A TransportID ACL entry for that initiator that includes an accessible logical unit pair with LUN value matching the addressed LUN.
b) The initiator is in the enrolled state (see 0.0.14.3) under an AccessID and that AccessID has an ACL entry that includes an accessible logical unit pair with LUN value matching the addressed LUN.
c) The addressed LUN matches a LUN value assigned via a valid proxy token via the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action.

If the initiator has access to the logical unit by virtue of an AccessID enrolled by that initiator and the initiator is in the pending-enrolled state, then commands shall be processed as follows:

a) INQUIRY, REPORT LUNS, ACCESS CONTROL OUT and ACCESS CONTROL IN shall be processed as if access controls were not present;
b) all other commands shall be terminated prior to any data transfer with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INITIATOR PENDING-ENROLLED.

NOTE An initiator should respond to the ACCESS DENIED - INITIATOR PENDING-ENROLLED additional sense code by sending an ACCESS CONTROL OUT command with ACCESS ID ENROLL service action. If the command succeeds, the initiator may retry the failed command.

If an INQUIRY command is addressed to a LUN that is not associated for that initiator to an accessible logical unit, the device server shall set the Peripheral Device Type to 1Fh and Peripheral Qualifier to 011b (the device server is not capable of supporting a device at this logical unit).

The parameter data returned in response to a REPORT LUNS command addressed to LUN 0 shall return only the list of LUN values that are associated to accessible logical units. If the initiator is in the enrolled or pending-enrolled state, this list shall include any LUN values associated to accessible logical units by virtue of the AccessID enrolled by that initiator. If the initiator (in any enrollment state) has access to any logical units by virtue of proxy tokens, the corresponding LUN values are also included in the parameter data. If the initiator (in the not-enrolled state) has no access rights to any logical unit (either through a TransportID or through a Proxy Token), then the response to REPORT LUNS shall include only LUN 0, as specified in SPC-3 rev 0, 7.19.

**AUTHOR'S NOTE:** *The reference above will need to be checked after this clause is inserted into SPC-3.*

Except when access controls are disabled, all cases not described previously in this subclause shall result in termination of the command with CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to LOGICAL UNIT NOT SUPPORTED.

### 4.11 Access Controls Log

The access controls log is a record of events related to the access controls state.

The log has three portions recording different classes of events:

a) key override events (when an attempt is made to override the Management Identifier Key, whether the attempt fails or succeeds);
b) invalid key events (when the Management Identifier Key in a CDB or parameter data does not match the current value maintained by the access controls coordinator);
c) ACL LUN conflict events (see 0.0.15).

Each portion of the log is required to contain a counter of the events. When a device ships from the factory, the counters shall be zero. The counters are increased by one whenever the relevant event occurs. Optionally, each log portion may contain additional records with more specific information about each event as described in the following paragraphs of this clause.

The override key events occur when the access controls coordinator receives the ACCESS CONTROL OUT command with OVERRIDE MGMT KEY service action. When such an event occurs, the access controls coordinator shall increase the Key Overrides Counter by one. If the log has additional resources to record event details, the access controls coordinator shall prepend to this portion of the log a record that includes the TransportID of the initiator that sent the command, a flag that indicates if the override was successful, the current value and initial setting of the Override Lockout Timer, and a 32 bit integer time-stamp. (See 0.0.23.2.)

The invalid key events occur whenever an access controls command requires checking a field either in the CDB or in the parameter data against the current Management Identifier Key and this check fails because the value in the field does not equal the current value maintained by the access controls coordinator. When such an event occurs, the access controls coordinator shall increase the Invalid Keys Counter by one. If the log has additional resources to record event details, the access controls coordinator shall prepend to this portion of the log a record that includes the TransportID of the initiator that sent the command, the operation code of the command and its service action, the invalid key and a 32 bit integer time-stamp. (See 0.0.23.3.)

The ACL LUN conflict events occur as specified in 0.0.15. When such an event occurs, the access controls coordinator shall increase the ACL LUN Conflicts Counter by one. If the log has additional resources to record event details, the access controls coordinator shall prepend to this portion of the log a record that includes the TransportID of the initiator enrolling the AccessID that created the conflict, and a 32 bit integer time-stamp. (See 0.0.23.4.)

The content of the time stamp fields described in the log records are vendor specific. If the device has no time stamp resources the fields 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.

If the additional event records resources are exhausted, new records are always prepended to the log and the oldest records are deleted.

Selected portions of the log may be reported to an application client by the ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action (see 0.0.23). With the exception of the key overrides portion, selected portions of the log may be cleared and the counters reset to zero with the ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action (see 0.0.31).

## ~~4.2 Resource requirements for Access Controls~~

~~If a device supports the access controls, then the device shall contain an access controls coordinator that shall be able to maintain the following data structures:~~ The information shown in table t5 shall be maintained by the access controls coordinator.

**Table t5 — Mandatory access controls resources**

| Information Description | Size (in bits) | Persistent Across Power Cycles and Resets |
|---|---|---|
| One ACL (see 5.99.2)<br>containing at least one ACE<br>containing<br>one access identifier (see 5.99.2.2); and<br>at least one LUACD (see 5.99.2.3) | VS | Yes |
| The Enrollment State for each initiator (see 5.99.4.1) | VS | Yes |
| Management Identifier Key (see 5.99.7) | 64 | Yes |
| Default LUNs Generation a.k.a. DLgeneration (see 5.99.3.4) | 32 | Yes |
| Override Lockout Timer (see 5.99.7.2.2) | 16 | No |
| Initial Override Lockout Timer value (see 5.99.7.2.2) | 16 | Yes |
| Access Controls Log Event Counters (see 5.99.9) | | Yes |
| containing at least the following: | | |
| Key Overrides Counter | 16 | Yes |
| Invalid Keys Counter | 16 | Yes |
| ACL LUN Conflicts Counter | 16 | Yes |

- ~~a) An ACL consisting of at least one entry where each entry shall contain at least one accessible logical unit pair;~~
- ~~b) an 8-byte (64 bit) integer called the Management Identifier Key (see x.x.x and 0.0.27);~~
- ~~c) a 4-byte (32 bit) integer called the Default LUNs Generation (see x.x.x);~~
- ~~d) a 2-byte (16 bit) integer called the Initial Override Lockout Timer (see x.x.x);~~
- ~~e) a log of access controls related events containing at least the following (see x.x.x):~~
  - ~~A) a 2-byte (16 bit) integer called the Key Overrides Counter;~~
  - ~~B) a 2-byte (16 bit) integer called the Invalid Keys Counter;~~
  - ~~C) a 2-byte (16 bit) integer called the ACL LUN Conflicts Counter.~~

Optionally, the access controls coordinator may maintain ~~additional data structures to manage proxy tokens for some or all of the device's logical units (see 0.0.19)~~ the information shown in table t6.

**Table t6 — Optional access controls resources**

| **Information Description** | **Size**<br>(in bits) | **Persistent Across Power Cycles and Resets** |
|---|---|---|
| One or more proxy tokens (see 5.99.5.2.1) | 64 | Yes |
| One or more proxy LUNs (see 5.99.5.2.2) | 64 | No |
| Access controls log event records (see 5.99.9) for | | |
| Key Overrides events | (see 7.1.4.2.2) | Yes |
| Invalid Keys events | (see 7.1.4.2.3) | Yes |
| ACL LUN Conflicts events | (see 7.1.4.2.4) | Yes |

When shipped from the factory, the ACL ~~is~~ shall be empty, all ~~integer~~ values ~~are~~ shown in table t5 shall be zero, additional access control log structures ~~are~~ shall be empty and there ~~are~~ shall be no valid proxy tokens.

~~Persistence of these data structures through power-cycles or target resets is described in x.x.x.~~

## F.7 – SPC-3 Command Definitions

Clause 7 of SPC-3 should have the following command definitions added.

## 7.1 ACCESS CONTROL IN command

### 7.1.1 ACCESS CONTROL IN introduction ~~command descriptor block~~

The service actions of the ACCESS CONTROL IN command (see table t7) ~~is~~ are used to obtain information about the access controls that are active within the access controls coordinator and to facilitate other access control functions (see 5.99). ~~The command shall be used in conjunction with~~ If the ACCESS CONTROL IN command is implemented, the ACCESS CONTROL OUT command also shall be implemented. The ACCESS CONTROL IN command ~~It~~ shall not be affected by ~~reservations, persistent reservations or~~ access controls.

**Table t7 — ACCESS CONTROL IN service actions**

| Service Action | Command name | Type | Reference |
|---|---|---|---|
| 00h | REPORT ACL | m | 7.1.2 |
| 01h | REPORT LU DESCRIPTORS | m | 7.1.3 |
| 02h | REPORT ACCESS CONTROLS LOG | m | 7.1.4 |
| 03h | REPORT OVERRIDE LOCKOUT TIMER | m | 7.1.5 |
| 04h | REQUEST PROXY TOKEN | o | 7.1.6 |
| 05h - 17h | Reserved | | |
| 18h - 1Fh | Vendor specific | | |
| Key: m = Service action implementation is mandatory if ACCESS CONTROL IN is implemented.<br>o = Service action implementation is optional. | | | |

If the device contains an access controls coordinator, ~~this~~ the ACCESS CONTROL IN command shall be processed by the access controls coordinator if addressed to LUN 0. ~~or~~ The ACCESS CONTROL IN command also may be addressed to any other LUN value whose standard INQUIRY data (see 7.z.z) has the ACC bit set to one, in which case it~~. In the latter case, the command~~ shall be processed in the same manner as if the command had been addressed to LUN 0. If an ACCESS CONTROL IN command is received by a device server whose standard INQUIRY data has the ACC bit set to zero, the command ~~It~~ shall be ~~rejected by the device server if addressed to any other LUN~~ 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 ~~OPCODE~~.

~~Table 6: ACCESS CONTROL IN command~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~operation code (86h)~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | ~~service action~~ | | | | |
| ~~2~~<br>~~9~~ | ~~MSB~~ | ~~service action-specific data~~ | | | | | | ~~LSB~~ |
| ~~10~~<br>~~13~~ | ~~MSB~~ | ~~service action-specific data2 or allocation length~~ | | | | | | ~~LSB~~ |
| ~~14~~ | ~~Reserved~~ | | | | | | | |
| ~~15~~ | ~~control~~ | | | | | | | |

~~The SERVICE ACTION-SPECIFIC DATA field is described in the appropriate subclause for each service action.~~

~~The SERVICE ACTION-SPECIFIED DATA2 field or the ALLOCATION LENGTH field are distinguished in the appropriate subclause for each service action. When the field is interpreted as an Allocation Length, the ALLOCATION LENGTH field shall conform to the requirements of clause 4.3.4.6 (of SPC-3 revision 0).~~

~~The actual length of the ACCESS CONTROL IN parameter list is available in or may be derived from a parameter list field in those cases where the parameter data has variable length.~~

## ~~5.2 ACCESS CONTROL IN Service Actions~~

### ~~5.2.1 ACCESS CONTROL IN Service Action Codes~~

~~Table 6 gives a summary of the ACCESS CONTROL IN command service action codes.~~

~~Table 7: ACCESS CONTROL IN command service action codes (M=Mandatory, O=Optional, V=Vendor-specific)~~

| ~~Code~~ | ~~Name~~ | ~~Type~~ | ~~KeyRq~~ | ~~Clause~~ |
|---|---|---|---|---|
| ~~00h~~ | ~~REPORT ACL~~ | ~~M~~ | ~~Y~~ | ~~0.0.21~~ |
| ~~01h~~ | ~~REPORT LU DESCRIPTORS~~ | ~~M~~ | ~~Y~~ | ~~0.0.22~~ |
| ~~02h~~ | ~~REPORT ACCESS CONTROLS LOG~~ | ~~M~~ | ~~Y~~ | ~~0.0.23~~ |
| ~~03h~~ | ~~REPORT OVERRIDE LOCKOUT TIMER~~ | ~~M~~ | ~~Y~~ | ~~0.0.24~~ |
| ~~04h~~ | ~~REQUEST PROXY TOKEN~~ | ~~O~~ | ~~N~~ | ~~0.0.25~~ |
| ~~05h-17h~~ | ~~Reserved~~ | | | |
| ~~18h-1Fh~~ | ~~Vendor-specific~~ | ~~V~~ | | |

~~The KeyRq column indicates whether the Management Identifier Key shall be supplied for successful completion of the service action (with the exception of special cases where no data is transferred). A "Y" indicates that the Management Identifier Key is required. An "N" indicates that the Management Identifier Key is not required.~~

### 7.1.2 REPORT ACL service action ~~(Mandatory)~~

#### 7.1.2.1 REPORT ACL introduction ~~service action command descriptor block~~

The ACCESS CONTROL IN command with REPORT ACL service action (see table t8) ~~of the ACCESS CONTROL IN command~~ is used ~~by an application client~~ to query the ~~complete~~ ACL (see 5.99.2) ~~currently~~ maintained by the access controls coordinator. If the ACCESS CONTROL IN command is implemented, the REPORT ACL service action shall be implemented.

**Table t8 — ACCESS CONTROL IN command with REPORT ACL service action**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (86h) | | | | | | | |
| 1 | Reserved | | | | SERVICE ACTION (00h) | | | |
| 2 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | (MSB) | ALLOCATION LENGTH | | | | | | |
| 13 | | | | | | | | (LSB) |
| 14 | Reserved | | | | | | | |
| 15 | CONTROL | | | | | | | |

If access controls are disabled, the device server shall ignore the MANAGEMENT IDENTIFIER KEY field and shall respond with GOOD status ~~and~~ returning only the eight ~~(8)~~ byte parameter list header ~~as~~ specified in 7.1.2.2 subject to the ALLOCATION LENGTH limitation described in 4.3.4.6~~, regardless of the value of any other field in the CDB~~.

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not match~~, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain~~ the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, parameter data shall not be returned, the command shall be terminated~~. If this is not the case, the device server shall return no data and respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense ~~data~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9).

~~If access controls are enabled and the SERVICE ACTION-SPECIFIC DATA field in the CDB matches the current Management Identifier Key maintained by the access controls coordinator, then the format of the returned data shall conform to the specification in 0.0.21.2.~~

The ALLOCATION LENGTH field is described in 4.3.4.6. The ALLOCATION LENGTH field value should be at least eight~~ (8), sufficient for the header information~~.

### 7.1.2.2 REPORT ACL parameter data format

#### 7.1.2.2.1 REPORT ACL parameter data introduction ~~header~~

The format of the parameter data ~~provided~~ returned in response to an ACCESS CONTROL IN command with REPORT ACL service actions is shown in table t9. ~~The ACL Entry Page(s) are described in 0.0.21.2.2 and~~

Table t9 — ACCESS CONTROL IN with REPORT ACL parameter data format

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Parameter list header | | | | | | | |
| 0 | (MSB) | ~~additional~~ ACL DATA LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | DLGENERATION ~~default luns generation~~ | | | | | | |
| 7 | | | | | | | | (LSB) |
| | ~~ACL entry~~ ACL data pages | | | | | | | |
| 8 | ACL data page 0 | | | | | | | |
| | | | | | | | | |
| | ⋮ | | | | | | | |
| | | | | | | | | |
| n | ACL data page x | | | | | | | |

~~0.0.21.2.3.~~

~~Table 8: REPORT ACL parameter data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~MSB~~ | ~~additional length (*n*-3)~~ | | | | | | |
| ~~3~~ | | | | | | | | ~~LSB~~ |
| ~~4~~ | ~~MSB~~ | ~~default luns generation~~ | | | | | | |
| ~~7~~ | | | | | | | | ~~LSB~~ |
| ~~8~~<br>~~n~~ | ~~ACL Entry Page(s)~~ | | | | | | | |

The ~~ADDITIONAL~~ ACL DATA LENGTH field shall contain a count of the number of bytes in the remaining parameter data. The value in ~~this~~ the ACL DATA LENGTH field shall be ~~contain~~ the actual number of bytes available without consideration for insufficient allocation length in the ~~requesting~~ CDB. If access controls are disabled, the ~~ADDITIONAL~~ ACL DATA LENGTH field ~~in the returned parameter data~~ shall be set to four ~~(4)~~.

The DLGENERATION ~~DEFAULT LUNS GENERATION~~ field shall contain ~~be set to~~ the current DLgeneration value ~~of the Default LUNs Generation integer maintained by the access controls coordinator according to the rules in~~ (see 5.99.3.4).

The ~~ACL Entry~~ ACL data pages~~(s) shall~~ contain a description of the ACL (see 5.99.2) maintained by the access controls coordinator. Each ~~ACL Entry~~ ACL data page describes one ACE in the ACL or one proxy token (see

5.99.5.2). Every ACE and every proxy token managed by the access controls coordinator shall have an ACL data page in the parameter data. The content and format of an ACL data page is indicated ~~identified~~ by a page code (see table t10).

**Table t10 — ACL data page codes**

| Page Code | Description | Reference |
|---|---|---|
| 00h | Granted | 7.1.2.2.2 |
| 01h | Granted All | 7.1.2.2.3 |
| 02h | Proxy Tokens | 7.1.2.2.4 |
| 03h-EFh | Reserved | |
| F0h-FFh | Vendor specific | |

~~The list of Page Codes and their definitions is given in Table 8 and the detailed description of the pages are in subsequent subclauses.~~

~~TABLE 9: ACL Entry PAGE CODE definitions for REPORT ACL service action~~

| ~~Page Code~~ | ~~Description~~ | ~~Clause~~ |
|---|---|---|
| ~~00h~~ | ~~Granted~~ | ~~0.0.21.2.2~~ |
| ~~01h~~ | ~~Granted All~~ | ~~0.0.21.2.2~~ |
| ~~02h~~ | ~~Proxy Tokens~~ | ~~0.0.21.2.3~~ |
| ~~03h-EFh~~ | ~~Reserved~~ | |
| ~~F0h-FFh~~ | ~~Vendor-specific~~ | |

### 7.1.2.2.2 ~~REPORT ACL parameter data~~ Granted ~~and Granted All~~ ACL data page format

The Granted ACL data page (see table t11) describes an ACE that allows access to a specific set of logical units via a list of LUACDs (see 5.99.2.3).

**Table t11 — Granted ACL data page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (00h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | PAGE LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | Reserved | | | | | | | |
| 5 | ACCESS IDENTIFIER TYPE | | | | | | | |
| 6 | (MSB) | ACCESS IDENTIFIER LENGTH (m-7) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | ACCESS IDENTIFIER | | | | | | | |
| m | | | | | | | | |
| | LUACD Descriptors | | | | | | | |
| m+1 | LUACD descriptor 0 | | | | | | | |
| m+20 | | | | | | | | |
| | ⋮ | | | | | | | |
| n-19 | LUACD descriptor x | | | | | | | |
| n | | | | | | | | |

The PAGE LENGTH field shall indicate the number of additional bytes required for this page and shall not be adjusted to reflect any truncation caused by insufficient allocation length.

~~The identifier type and access identifier fields are specified in 7.1. The identifier length field indicates the number of bytes following taken up by the access identifier field.~~

The ACCESS IDENTIFIER TYPE field (see table t12) ~~is specified in 0.0.0.3~~ indicates the format and usage of the access identifier.

**Table t12 — Access Identifier types**

| Access Identifier Type | Access Identifier Name | Access Identifier Format Reference |
|---|---|---|
| 00h | AccessID | 5.99.2.2.2 |
| 01h | TransportID | 8.99.99.3 |
| 02h-7Fh | Reserved | |
| 80h-FFh | Vendor specific | |

The ACCESS IDENTIFIER LENGTH field indicates the number of bytes following taken up by the ACCESS IDENTIFIER field.

The ACCESS IDENTIFIER field ~~is specified in 7.x.y.~~ contains the identifier that the access controls coordinator uses to select the initiator(s) that are allowed access to the logical units named by the LUACD descriptors in this ACL data page. The format of the ACCESS IDENTIFIER field is specified in table t12. One and only one Granted or Granted All (see 7.1.2.2.3) page shall be returned for a given value in the ACCESS IDENTIFIER field.

~~NOTE All currently defined Identifier Types require the IDENTIFIER LENGTH field be set to 24 (see Table 35).~~

Each LUACD descriptor (see table t13) describes the access allowed to one logical unit based on the access identifier. There shall be one LUACD descriptor for each logical unit to which the access identifier allows access.

**Table t13 — Granted ACL page LUACD descriptor format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ACCESS MODE | | | | | | | |
| 1 | Reserved | | | | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | LUN VALUE | | | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | (MSB) | DEFAULT LUN | | | | | | |
| 19 | | | | | | | | (LSB) |

~~In each block, the value in~~ The ACCESS MODE field (see table t14) ~~shall~~ indicates the type of access allowed ~~that the specified initiator has~~ to the logical unit referenced by the DEFAULT LUN ~~value~~ field and addressable at the specified LUN value.

**Table t14 — Access mode values**

| Access Mode | Description |
|---|---|
| 00h | Normal access |
| 01h-EFh | Reserved |
| F0h-FFh | Vendor-specific |

The LUN VALUE field indicates the LUN value an accessing initiator would use to access the logical unit to which the LUACD descriptor applies.

The DEFAULT LUN field identifies the logical unit to which access is allowed using the default LUN value described in 5.99.3.3. The value in the DEFAULT LUN field shall be consistent with the DLGENERATION field contents returned in the parameter list header (see 7.1.2.2).

NOTE 3 - It is acceptable for the LUN VALUE and DEFAULT LUN fields to contain the same value.

~~The value 00h for ACCESS MODE shall mean normal access. The meaning of the values F0h-FFh are vendor-specific. The values 01h-EFh are reserved.~~

~~The Granted and Granted All page formats for the REPORT ACL service action are specified in Table 7. A Granted or Granted All page is used to report one entry in the ACL.~~

~~Table 10: ACL Entry Page: Granted and Granted Default page formats~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~page code (00h-01h)~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~MSB~~ | ~~page length (*m*-3)~~ | | | | | | ~~LSB~~ |
| ~~4~~ | ~~Reserved~~ | | | | | | | |
| ~~5~~ | ~~identifier type~~ | | | | | | | |
| ~~6~~<br>~~7~~ | ~~MSB~~ | ~~identifier length (*n*-7)~~ | | | | | | ~~LSB~~ |
| ~~8~~<br>~~n~~ | ~~MSB~~ | ~~access identifier~~ | | | | | | ~~LSB~~ |
| ~~n+1~~<br>~~m~~ | ~~lun/default lun list~~ | | | | | | | |

~~The LUN/DEFAULT LUN LIST field shall contain a list of LUN/default LUN pairs as specified in Table 10 that describe the accessible logical unit pairs in the ACL entry for the specified access identifier. The default LUN values in these pairs shall be consistent with the Default LUNs Generation value in the header of the parameter data.~~

~~Table 11: LUN/DEFAULT LUN LIST format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~access mode~~ | | | | | | | |
| ~~1~~<br>~~3~~ | ~~Reserved~~ | | | | | | | |
| ~~4~~<br>~~11~~ | ~~MSB~~ | ~~first lun~~ | | | | | | ~~LSB~~ |
| ~~12~~<br>~~19~~ | ~~MSB~~ | ~~first default lun~~ | | | | | | ~~LSB~~ |
| | ⋮ | | | | | | | |
| ~~n-19~~ | ~~access mode~~ | | | | | | | |
| ~~n-18~~<br>~~n-16~~ | ~~Reserved~~ | | | | | | | |
| ~~n-15~~<br>~~n-8~~ | ~~MSB~~ | ~~last lun~~ | | | | | | ~~LSB~~ |
| ~~n-7~~<br>~~n~~ | ~~MSB~~ | ~~last default lun~~ | | | | | | ~~LSB~~ |

~~In each block, the value in the ACCESS MODE field shall indicate the type of access that the specified initiator has to the logical unit referenced by the DEFAULT LUN value and addressable at the specified LUN value. The value 00h for~~

~~ACCESS MODE shall mean normal access. The meaning of the values F0h-FFh are vendor-specific. The values 01h-EFh are reserved.~~

~~For the Granted All page, the LUN/DEFAULT LUN LIST field is empty.~~

~~If an ACL entry for a specific access identifier has an accessible logical unit pairs list that does not contain a pair for every logical unit or for any pair the LUN value does not equal the default LUN value for the referenced logical unit, then the access controls coordinator shall include one Granted page for that access identifier and shall include in this page a complete list of LUN/default LUN pairs describing the list of accessible logical unit pairs in the ACL entry for that access identifier.~~

~~If an ACL entry for a specific access identifier has an accessible logical unit pairs list that contains a pair for every logical unit and each pair has LUN value equal to the default LUN value for the referenced logical unit, then the access controls coordinator shall include either one Granted All page or one Granted page for that access identifier. In the latter case, the Granted page shall contain a complete list of LUN/default LUN pairs for all logical units (with LUN value equal to the default LUN value in each pair).~~

~~One and only one Granted or Granted All page shall be returned for a given value in the ACCESS IDENTIFIER field.~~

### 7.1.2.2.3 Granted All ACL data page format

The Granted All ACL data page (see table t15) describes an ACE that allows access to all the SCSI target device's logical units with the default LUN value as the accessing LUN value. When an access identifier is present in a Granted All ACL data page, initiators that access via that access identifier are allowed to access the SCSI target device as if access controls were disabled.

**Table t15 — Granted All ACL data page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (01h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | PAGE LENGTH (m-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | Reserved | | | | | | | |
| 5 | ACCESS IDENTIFIER TYPE | | | | | | | |
| 6 | (MSB) | ACCESS IDENTIFIER LENGTH (m-7) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | ACCESS IDENTIFIER | | | | | | | |
| m | | | | | | | | |

The PAGE LENGTH, ACCESS IDENTIFIER TYPE, and ACCESS IDENTIFIER LENGTH, are described in 7.1.2.2.2.

The ACCESS IDENTIFIER field contains the identifier that the access controls coordinator uses to select the initiator(s) that are allowed access to all the SCSI target device's logical units with the default LUN value as the accessing LUN value. The format of the access identifier field is specified in table t12 (see 7.1.2.2.2). One and only one Granted (see 7.1.2.2.2) or Granted All page shall be returned for a given value in the ACCESS IDENTIFIER field.

### 7.1.2.2.4 ~~REPORT ACL parameter data~~ Proxy tokens ACL data page format

~~The Proxy Tokens page format for the REPORT ACL service action is specified in Table 9.~~

~~Table 12: ACL Entry Page: Proxy Tokens page format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~page code (02h)~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~page length (m-3)~~ | | | | | | | |
| ~~4~~<br>~~m~~ | ~~proxy token/default lun list~~ | | | | | | | |

The proxy tokens page (see table t16) describes the proxy tokens (see 5.99.5.2) maintained by the access controls coordinator.

**Table t16 — Proxy tokens data page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (02h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | PAGE LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| | Proxy token Descriptors | | | | | | | |
| 4 | Proxy token descriptor 0 | | | | | | | |
| 23 | | | | | | | | |
| | .<br>. | | | | | | | |
| n-19 | Proxy token descriptor x | | | | | | | |
| n | | | | | | | | |

The PAGE LENGTH field shall indicate the number of additional bytes required for this page and shall not be adjusted to reflect any truncation caused by insufficient allocation length.

If there are no active proxy tokens ~~at the access controls coordinator~~, the access controls coordinator may either not include the proxy tokens page in the parameter data or may include one such page containing no proxy token descriptors ~~with an empty PROXY TOKEN/DEFAULT LUN LIST field~~.

At most one proxy tokens page shall be included in the parameter data.

Each proxy token descriptor (see table t17) describes the access allowed to one logical unit based on one proxy token. There shall be one proxy token descriptor for each active proxy token maintained by the access controls coordinator.

**Table t17 — Proxy token descriptor format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | Reserved | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | | | PROXY TOKEN | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | (MSB) | | | DEFAULT LUN | | | | |
| 19 | | | | | | | | (LSB) |

The PROXY TOKEN field indicates the proxy token to which this proxy token descriptor applies.

The DEFAULT LUN field identifies the logical unit to which this proxy token allows access using the default LUN value described in 5.99.3.3. The value in the DEFAULT LUN field shall be consistent with the DLGENERATION value returned in the parameter list header (see 7.1.2.2).

NOTE 4 - The same default LUN value may appear in multiple proxy token descriptors, if multiple proxy tokens are valid for the same logical unit.

~~The PROXY TOKEN/DEFAULT LUN LIST field shall contain a list of Proxy Token/default LUN pairs as specified in Table 12 indicating the association of Proxy Token to logical unit. The default LUN values in these pairs shall be consistent with the Default LUNs Generation value in the header of the parameter data.~~

~~Table 13: PROXY TOKEN/DEFAULT LUN LIST format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~3~~ | | | | ~~Reserved~~ | | | | |
| ~~4~~<br>~~11~~ | ~~MSB~~ | | | ~~first proxy token~~ | | | | ~~LSB~~ |
| ~~12~~<br>~~19~~ | ~~MSB~~ | | | ~~first default lun~~ | | | | ~~LSB~~ |
| | | | | ~~.~~<br>~~.~~<br>~~.~~ | | | | |
| ~~*n*-19~~<br>~~*n*-16~~ | | | | ~~Reserved~~ | | | | |
| ~~*n*-15~~<br>~~*n*-8~~ | ~~MSB~~ | | | ~~last proxy token~~ | | | | ~~LSB~~ |
| ~~*n*-7~~<br>~~*n*~~ | ~~MSB~~ | | | ~~last default lun~~ | | | | ~~LSB~~ |

~~There may be multiple Proxy Token/default LUN pairs with the same default LUN value if multiple proxy tokens are valid for the same logical unit.~~

### 7.1.3 REPORT LU DESCRIPTORS service action ~~(Mandatory)~~

#### 7.1.3.1 REPORT LU DESCRIPTORS introduction ~~service action command descriptor block~~

The ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action (see table t18) ~~of the ACCESS CONTROL IN command~~ is used ~~by an application client~~ to obtain ~~from~~ the ~~access controls coordinator~~ inventory of ~~information about the~~ logical units for which access controls may be established ~~and other properties of the access controls coordinator~~. If the ACCESS CONTROL IN command is implemented, the REPORT LU DESCRIPTORS service action shall be implemented.

**Table t18 — ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (86h) | | | | | | | |
| 1 | Reserved | | | SERVICE ACTION (01h) | | | | |
| 2 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | (MSB) | ALLOCATION LENGTH | | | | | | |
| 13 | | | | | | | | (LSB) |
| 14 | Reserved | | | | | | | |
| 15 | CONTROL | | | | | | | |

If access controls are disabled, the device server shall ignore the MANAGEMENT IDENTIFIER KEY field and shall respond with GOOD status ~~and~~ returning only the twenty ~~(20)~~ byte parameter list header as specified in  subject to the ALLOCATION LENGTH limitation described in 4.3.4.6~~, regardless of the value of any other field in the CDB.~~

NOTE 5 - When access controls are disabled ~~In this case~~, the logical unit inventory ~~all logical units are accessible to all initiators; existing~~ may be obtained using commands such as ~~INQUIRY,~~ REPORT LUNS (see 7.z.z)~~, READ CAPACITY, etc., may be used to collect this information if needed~~. To facilitate access controls management the ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action returns more information than the REPORT LUNS command. When access controls are disabled additional commands such as INQUIRY (see 7.z.z) are require to obtain all the information provided by the ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action.

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not match~~, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain~~ the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, parameter data shall not be returned, the command shall be terminated~~. If this is not the case, the device server shall return no data and respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense ~~data~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9).

~~If access controls are enabled and the SERVICE ACTION-SPECIFIC DATA field in the CDB matches the current Management Identifier Key maintained by the access controls coordinator, then the format of the returned data shall conform to the specification in 0.0.22.2.~~

The ALLOCATION LENGTH field is described in 4.3.4.6. The ALLOCATION LENGTH field value should be at least twenty~~-(20), sufficient for the header information~~.

### 7.1.3.2 REPORT LU DESCRIPTORS parameter data format

#### ~~7.1.3.2.1 REPORT LU DESCRIPTORS parameter data header~~

The format of the parameter data ~~provided~~ returned in response to an ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service actions is shown in table t19.

**Table t19 — ACCESS CONTROL IN with REPORT LU DESCRIPTORS parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Parameter list header | | | | | | | |
| 0 | (MSB) | | | ~~additional~~ LU INVENTORY LENGTH (n-3) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | NUMBER OF LOGICAL UNITS | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | | | SUPPORTED LUN-MASK FORMAT | | | | |
| 15 | | | | | | | | (LSB) |
| 16 | (MSB) | | | DLGENERATION ~~default luns generation~~ | | | | |
| 19 | | | | | | | | (LSB) |
| | Logical Unit descriptors | | | | | | | |
| 20 | | | | Logical Unit descriptor 0 | | | | |
| | | | | | | | | |
| | : | | | | | | | |
| | | | | Logical Unit descriptor x | | | | |
| n | | | | | | | | |

### ~~5.2.3.2.1 REPORT LU DESCRIPTORS parameter data, all devices~~

~~The format for the parameter data provided in response to an ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action is shown in Table 13.~~

~~Table 14: REPORT LU DESCRIPTORS parameter data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~3~~ | ~~MSB~~ | | | ~~additional length (*n*-3)~~ | | | | ~~LSB~~ |
| ~~4~~<br>~~7~~ | ~~MSB~~ | | | ~~number of logical units~~ | | | | ~~LSB~~ |
| ~~8~~<br>~~15~~ | | | | ~~supported lun-mask format~~ | | | | |
| ~~16~~<br>~~19~~ | ~~MSB~~ | | | ~~default luns generation~~ | | | | ~~LSB~~ |
| ~~20~~<br>~~n~~ | | | | ~~logical unit descriptors~~ | | | | |

The ~~ADDITIONAL~~ LU INVENTORY LENGTH field shall contain a count of the number of bytes in the remaining parameter data. The value in ~~this~~ the LU INVENTORY LENGTH field shall be ~~contain~~ the actual number of bytes available without consideration for insufficient allocation length in the CDB. If access controls are disabled, the ~~ADDITIONAL~~ LU INVENTORY LENGTH field shall be set to sixteen ~~(16)~~.

The NUMBER OF LOGICAL UNITS field shall contain a count of the number of logical units managed by the access controls coordinator. ~~(this~~ The value in NUMBER OF LOGICAL UNITS field shall be the same as the number of ~~LOGICAL UNIT DESCRIPTORS~~ Logical Unit descriptors that follow in the ~~remaining~~ parameter data~~)~~.

The SUPPORTED LUN-MASK FORMAT field (see table t20) contains a summary of the LUN values (see 5.99.2.3) that the access controls coordinator supports~~ in an accessible logical unit pair in an ACL entry. The format is specified in Table 14~~. LUN values are exchanged between application clients and the access controls coordinator by several service actions (e.g., the REPORT ACL IN command with REPORT ACL service action described in 7.1.2 and the REPORT ACL OUT command with MANAGE ACL service action described in 7.x.y). The format of the SUPPORTED LUN-MASK FORMAT field follows the eight byte LUN structure defined for dependent logical units by SAM-2.

**Table t20 — SUPPORTED LUN-MASK FORMAT field format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | | | FIRST LEVEL LUN MASK ~~part one~~ | | | | |
| 1 | | | | | | | | (LSB) |
| 2 | (MSB) | | | SECOND LEVEL LUN MASK ~~part two~~ | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | | | THIRD LEVEL LUN MASK ~~part three~~ | | | | |
| 5 | | | | | | | | (LSB) |
| 6 | (MSB) | | | FOURTH LEVEL LUN MASK ~~part four~~ | | | | |
| 7 | | | | | | | | (LSB) |

~~Table 15: SUPPORTED LUN-MASK FORMAT data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~MSB~~ | | | | | | | |
| ~~1~~ | | | | ~~lun-mask part one~~ | | | | ~~LSB~~ |
| ~~2~~ | ~~MSB~~ | | | | | | | |
| ~~3~~ | | | | ~~lun-mask part two~~ | | | | ~~LSB~~ |
| ~~4~~ | ~~MSB~~ | | | | | | | |
| ~~5~~ | | | | ~~lun-mask part three~~ | | | | ~~LSB~~ |
| ~~6~~ | ~~MSB~~ | | | | | | | |
| ~~7~~ | | | | ~~lun-mask part four~~ | | | | ~~LSB~~ |

~~Each of the four 2-byte fields specifies a mask of those bits that may be set within each field that the access controls coordinator supports for that portion of a LUN.~~

The LUN MASK at each level indicates approximately the logical unit number values the access controls coordinator supports. A bit value of zero in a LUN MASK field indicates that the access controls coordinator prohibits setting that bit to one in a LUN value. A bit value of one in a LUN MASK field indicates that the access controls coordinator may allow setting that bit to one in a LUN value.

For example, if the access controls coordinator ~~uses a flat addressing model and~~ only supports level one LUN values ~~at the top level~~ and up to 256 LUN values, then the SUPPORTED LUN-MASK FORMAT field shall contain 00FF000000000000h ~~LUN-MASK PART ONE field should be set to 255 (00FFh) and the LUN-MASK PART TWO, THREE and FOUR fields shall be set to zero~~. If only 200 LUN values were supported, the SUPPORTED LUN-MASK FORMAT field still would contain 00FF000000000000h.

~~The use of the mask format allows the access controls coordinator to suggest that it supports or simulates support for the hierarchical addressing model (see SAM-2).~~

~~NOTE~~ The value in the SUPPORT LUN-MASK FORMAT field ~~is intended~~ only ~~as a summary of~~ summarizes the supported LUN values and is not a complete description. The value in the SUPPORT LUN-MASK FORMAT field should be used as a guideline for specifying LUN values in service actions such as the ACCESS CONTROL OUT command with MANAGE ACL service action, it should not be viewed as a guarantee against rejection of requested LUN values. ~~It is possible that some bit combinations valid with respect to the SUPPORTED LUN-MASK FORMAT are not valid in practice.~~ ~~However, any bit combination inconsistent with the SUPPORTED LUN-MASK FORMAT shall not be valid.~~

The DLGENERATION ~~DEFAULT LUNS GENERATION~~ field shall contain ~~be set to~~ the current DLgeneration value ~~of the Default LUNs Generation integer maintained by the access controls coordinator according to the rules in~~ (see 5.99.3.4).

Each Logical Unit descriptor (see table t21) contains information about one logical unit managed by the access controls coordinator. There shall be one Logical Unit descriptor for every logical unit managed by the access controls coordinator.

**Table t21 — Logical Unit descriptor format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | PERIPHERAL DEVICE TYPE | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | ~~ADDITIONAL~~ DESCRIPTOR LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | DEFAULT LUN | | | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | Reserved | | | | | | | |
| 13 | EVPD IDENTIFICATION DESCRIPTOR LENGTH ~~(m)~~ | | | | | | | |
| 14 | Reserved | | | | | | | |
| 15 | DEVICE IDENTIFIER LENGTH ~~(k)~~ | | | | | | | |
| 16 | (MSB) | EVPD IDENTIFICATION DESCRIPTOR | | | | | | |
| 47 | | | | | | | | (LSB) |
| 48 | (MSB) | DEVICE IDENTIFIER | | | | | | |
| 79 | | | | | | | | (LSB) |
| 80 | (MSB) | DEVICE-TYPE SPECIFIC ~~ADDITIONAL~~ DATA | | | | | | |
| n | | | | | | | | (LSB) |

~~The LOGICAL UNIT DESCRIPTORS shall contain a description of the logical units managed by the access controls coordinator. Each descriptor is device-type specific but has the general format specified in Table 15.~~

~~Table 16: LOGICAL UNIT DESCRIPTOR data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~Reserved~~ | | | | ~~peripheral device-type~~ | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~MSB~~ | | | ~~additional length (*n*-3)~~ | | | | ~~LSB~~ |
| ~~4~~<br>~~11~~ | ~~MSB~~ | | | ~~default lun~~ | | | | ~~LSB~~ |
| ~~12~~ | ~~Reserved~~ | | | | | | | |
| ~~13~~ | ~~evpd identification descriptor length (*m*)~~ | | | | | | | |
| ~~14~~ | ~~Reserved~~ | | | | | | | |
| ~~15~~ | ~~device identifier length (*k*)~~ | | | | | | | |
| ~~16~~<br>~~47~~ | ~~MSB~~ | | | ~~evpd identification descriptor~~ | | | | ~~LSB~~ |
| ~~48~~<br>~~79~~ | ~~MSB~~ | | | ~~device identifier~~ | | | | ~~LSB~~ |
| ~~80~~<br>~~n~~ | ~~MSB~~ | | | ~~device-type specific additional data~~ | | | | ~~LSB~~ |

The PERIPHERAL DEVICE TYPE field is as defined in 7.z.z. ~~The PERIPHERAL DEVICE-TYPE field shall be set according to the device type of the referenced logical unit as specified in Table 48 (of SPC-3, rev 0).~~

The ~~ADDITIONAL~~ DESCRIPTOR LENGTH field indicates the total number of bytes remaining in the descriptor and shall not reflect any truncation of the parameter data as a result of insufficient allocation length. If the PERIPHERAL DEVICE TYPE field contains 0h, 4h, or 7h, the DESCRIPTOR LENGTH field shall contain 92 if the descriptor includes the DEVICE-TYPE SPECIFIC DATA field and 80 if it does not. If the PERIPHERAL DEVICE TYPE field contains any value other than 0h, 4h, or 7h, the DESCRIPTOR LENGTH field shall contain 80.

The DEFAULT LUN field contains the default LUN value (see 5.99.3.3) for the logical unit described by this logical unit descriptor. The value in the DEFAULT LUN field shall be consistent with the DLGENERATION value returned in the parameter list header (see 7.1.3.2).

~~The DEFAULT LUN field indicates the default LUN value associated to the referenced logical unit, as would be used in other commands (e.g., ACCESS CONTROL OUT command with MANAGE ACL service action) to identify the logical unit. This value shall be consistent with the Default LUNs Generation value in the header of the parameter data. This value shall be the same as would be returned in REPORT LUNS parameter data for the referenced logical unit if access controls were disabled.~~

The EVPD IDENTIFICATION DESCRIPTOR LENGTH field indicates the number of non pad bytes in the EVPD IDENTIFICATION DESCRIPTOR field.

The DEVICE IDENTIFIER LENGTH field indicated the number of non pad bytes in the DEVICE IDENTIFIER field.

The EVPD IDENTIFICATION DESCRIPTOR field shall contain non zero bytes ~~be supported~~ if:

a) The logical unit ~~device~~ supports the INQUIRY command (see 7.z.z) with EVPD bit set to one and the PAGE OR OPERATION CODE field ~~Page Code~~ set to 83h ~~(Device Identification Page)~~; and
b) At least one identification descriptor in the Device Identification VPD page (see 8.z.z) has 0h in the ASSOCIATION field ~~value of 0h (as defined in SPC-3 rev 0, 8.4.4, Table 173)~~.

Editors Note 1 - ROW: I believe that meeting both of the above requirements is now mandatory for all SPC-2 devices (and therefore for SPC-3 devices. So, I think the EVPD IDENTIFICATION DESCRIPTOR field must always contain non zero bytes.

When the above criteria are met ~~In this case~~, the EVPD IDENTIFICATION DESCRIPTOR field shall be derived from one of ~~these~~ the identification descriptors having 0h in the ASSOCIATION field as follows:

a) If the identification descriptor has a length less than or equal to ~~thirty-two (~~32~~)~~ bytes, then the EVPD IDENTIFICATION DESCRIPTOR field shall be set to the value of the identification descriptor in the most significant bytes of the field and the remainder of the field shall be padded with zero in the least significant bytes. ~~additionally,~~ The EVPD IDENTIFICATION DESCRIPTOR LENGTH field shall be set to the length of the identification descriptor; or
b) If the identification descriptor has a length greater than ~~thirty-two (~~32~~)~~ bytes, then the EVPD IDENTIFICATION DESCRIPTOR field shall be set to the ~~thirty-two (~~32~~)~~ most significant bytes of the identification descriptor. ~~additionally,~~ The EVPD IDENTIFICATION DESCRIPTOR LENGTH field shall be set to 32.

If there are several identification descriptors having 0h in the ASSOCIATION field, the choice of which descriptor to copy to the EVPD IDENTIFICATION DESCRIPTOR field is vendor specific, however, all ACCESS CONTROL IN commands with REPORT LU DESCRIPTORS service action shall return the same EVPD IDENTIFICATION DESCRIPTOR field contents for a given logical unit.

~~c) the same descriptor shall always be returned in this parameter data for the same logical unit; the choice of descriptor is vendor specific.~~

If no Device Identification VPD page identification descriptors with 0h in the ASSOCIATION field are available ~~If no such identification descriptor is available through INQUIRY~~, then the EVPD IDENTIFICATION DESCRIPTOR LENGTH field shall be set to zero and the EVPD IDENTIFICATION DESCRIPTOR field shall have all bytes set to zero.

~~The DEVICE IDENTIFIER field shall be supported if a device identifier has been established by a SET DEVICE IDENTIFIER command (see SPC-3, rev 0, 7.24). In this case, the DEVICE IDENTIFIER field shall be derived from this device identifier (what would be returned in response to a successful REPORT DEVICE IDENTIFIER command, see SPC-3, rev 0, 7.18) as follows:~~

If a device identifier has been set for the logical unit using the SET DEVICE IDENTIFIER command (see 7.z.z), the DEVICE IDENTIFIER field shall contain that device identifier subject to the following considerations:

a) If the device identifier has length less than or equal to ~~thirty-two (~~32~~)~~ bytes, then the DEVICE IDENTIFIER field shall be set to the value of the device identifier in the most significant bytes of the field and the remainder of the field shall be padded with zero in the least significant bytes. ~~additionally,~~ The DEVICE IDENTIFIER LENGTH field shall be set to the length of the device identifier; or
b) If the device identifier has length greater than ~~thirty-two (~~32~~)~~ bytes, then the DEVICE IDENTIFIER field shall be set to the ~~thirty-two (~~32~~)~~ most significant bytes of the identifier ~~descriptor additionally,~~ The DEVICE IDENTIFIER LENGTH field shall be set to 32.

If no ~~such~~ device identifier has been established by a SET DEVICE IDENTIFIER command, then the DEVICE IDENTIFIER LENGTH field shall be set to zero and the DEVICE IDENTIFIER field shall have all bytes set to zero.

~~**AUTHOR'S NOTE:** *the point of this truncation in both identifiers to 32 bytes is to reduce the amount of data that needs to be returned in this descriptor to manageable and consistent levels (we really don't want these logical unit descriptors to be arbitrarily large (device identifiers can be 2^32 bytes long!)). PAM probably doesn't need the full device identifier, just enough to help her keep track of devices. However, if need for more bytes from either identifier is required, an additional service action could be defined to request the complete information on an individual logical unit basis.*~~

If the PERIPHERAL DEVICE TYPE field contains any value other than 0h, 4h, or 7h, the DEVICE-TYPE SPECIFIC DATA field shall not be present in the Logical Unit descriptor.

~~The DEVICE-TYPE SPECIFIC ADDITIONAL DATA field for certain block devices is specified in 0.0.22.2.2. If clause 0.0.22.2.2 does not apply, this field shall not be included in the LOGICAL UNIT DESCRIPTOR, unless otherwise specified in the device-type specific command set standard applicable to the logical unit referenced by the LOGICAL UNIT DESCRIPTOR.~~

### ~~5.2.3.2.2 DEVICE-TYPE SPECIFIC ADDITIONAL DATA for certain block devices~~

The Logical Unit descriptor shall include the DEVICE-TYPE SPECIFIC DATA field if:

a) The PERIPHERAL DEVICE TYPE field contains 0h, 4h, or 7h;
b) The logical unit supports the READ CAPACITY command (see SBC-2) with:
    A) The RELADR bit set to zero; and
    B) The PMI bit set to zero; and
c) The logical unit standard INQUIRY data (see 7.z.z) has the RMB bit set to zero.

~~If the logical unit referenced by a Logical Unit Descriptor is a block device that~~

~~supports the READ CAPACITY command; and~~

~~the RMB bit in its Standard INQUIRY data indicates non-removable medium (RMB equal zero)~~

If the Logical Unit descriptor includes ~~then~~ the DEVICE-TYPE SPECIFIC ~~ADDITIONAL~~ DATA field ~~in the Logical Unit Descriptor~~, then the size of the DEVICE-TYPE SPECIFIC DATA field shall be ~~twelve (~~12~~)~~ bytes ~~long.~~ and the field shall contain ~~The data shall be~~ the same as the data that would be returned ~~for~~ by a successful READ CAPACITY command with LONGLBA bit set to one, and the RELADR and PMI bits set to zero.

~~**AUTHOR'S NOTE**: *the above sentence could use a cross-reference to SBC-2 (or someplace where the READ CAPACITY command, with the LONGLBA changes, is defined). Or we could reproduce that table here (see Table 7 of T10/99-259r4).*~~

### 7.1.4 REPORT ACCESS CONTROLS LOG service action ~~(Mandatory)~~

#### 7.1.4.1 REPORT ACCESS CONTROLS LOG introduction ~~service action command descriptor block~~

The ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action (see table t22) ~~of the ACCESS CONTROL IN command~~ is used ~~by an application client~~ to obtain ~~from~~ the ~~access controls coordinator~~ access controls log (see 5.99.9). If the ACCESS CONTROL IN command is implemented, the REPORT ACCESS CONTROLS LOG service action shall be implemented.

**Table t22 — ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (86h) | | | | | | | |
| 1 | Reserved | | | | SERVICE ACTION (02h) | | | |
| 2 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | Reserved | | | | | | LOG PORTION | |
| 11 | Reserved | | | | | | | |
| 12 | (MSB) | ALLOCATION LENGTH | | | | | | |
| 13 | | | | | | | | (LSB) |
| 14 | Reserved | | | | | | | |
| 15 | CONTROL | | | | | | | |

~~The SERVICE ACTION-SPECIFIC DATA2 field in the CDB shall have the structure specified in Table 16.~~

~~Table 17: REPORT ACCESS CONTROLS LOG SERVICE ACTION-SPECIFIC DATA2 field~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~Reserved~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | ~~log portion~~ | |
| ~~2~~ | ~~MSB~~ | ~~allocation length~~ | | | | | | |
| ~~3~~ | | | | | | | | ~~LSB~~ |

If access controls are disabled, the device server shall ignore the MANAGEMENT IDENTIFIER KEY field and shall respond with GOOD status ~~and~~ returning only the ~~four (4)~~ eight byte parameter list header as specified in 7.1.4.2.1 subject to the ALLOCATION LENGTH limitation described in 4.3.4.6~~, regardless of the value of any other field in the CDB.~~

If access controls are enabled and table t23 specifies that the management identifier key is not required then the device server shall ignore the contents of the MANAGEMENT IDENTIFIER KEY field.

If access controls are enabled, table t23 specifies that the management key identifier is required and the contents of the MANAGEMENT IDENTIFIER KEY field do not match~~, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain~~ the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, parameter data shall not be returned, the command shall be terminated~~. If this is not the case, the device server shall return no data and respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL

REQUEST, the additional sense ~~data~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9).

The LOG PORTION field (see table t23) ~~indicates to which~~ specifies the access controls log portion being requested ~~of the log this service action applies, as specified in Table 17~~.

**Table t23 — CDB LOG PORTION field values**

| Log Portion | Description | Management Identifier Key Required |
|---|---|---|
| 00b | Key Overrides portion | No |
| 01b | Invalid Keys portion | Yes |
| 10b | ACL LUN Conflicts portion | Yes |
| 11b | Reserved | |

~~TABLE 18: LOG PORTION field definitions for REPORT and CLEAR ACCESS CONTROLS LOG service actions~~

| ~~LOG PORTION~~ | ~~Description~~ | ~~Clause~~ |
|---|---|---|
| ~~00b~~ | ~~key overrides~~ | ~~0.0.23.2~~ |
| ~~01b~~ | ~~invalid keys~~ | ~~0.0.23.3~~ |
| ~~10b~~ | ~~ACL LUN conflicts~~ | ~~0.0.23.4~~ |
| ~~11b~~ | ~~Reserved~~ | |

The ALLOCATION LENGTH field is described in 4.3.4.6. The ALLOCATION LENGTH field value ~~in the SERVICE ACTION-SPECIFIC DATA2 field~~ should be at least eight ~~(8), sufficient for the header of the returned parameter data~~.

~~If the LOG PORTION field is set to 00b (key overrides), then the device server shall return in parameter data the contents of the key overrides portion of the log, as specified in 0.0.23.2, regardless of any other field in the CDB and regardless of whether access controls are enabled or disabled.~~

~~If the LOG PORTION field is set to any value other than 00b (key overrides) and if access controls are disabled, then the SERVICE ACTION-SPECIFIC DATA field shall be ignored and the device server shall return GOOD status and return only a four (4) byte header as specified in the relevant subclauses.~~

~~If the LOG PORTION field is set to any value other than 00b (key overrides) and if access controls are enabled, the following shall hold:~~

~~a) if the SERVICE ACTION-SPECIFIC DATA field in the CDB does not contain the current Management Identifier Key maintained by the access controls coordinator, then the device server shall return no data and respond with CHECK CONDITION, sense key ILLEGAL REQUEST, additional sense data set to ACCESS DENIED - INVALID MGMT ID KEY and the access controls coordinator shall record the event in the invalid keys portion of the access controls log (see a.b.c);~~

~~b) otherwise, the device server shall return in parameter data that portion of the log indicated in the LOG PORTION field, as specified in 0.0.23.3 and 0.0.23.4.~~

### 7.1.4.2 REPORT ACCESS CONTROLS LOG parameter data format

#### 7.1.4.2.1 REPORT ACCESS CONTROLS LOG parameter data introduction ~~header~~

The format of the parameter data returned in response to an ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service actions is shown in table t24.

**Table t24 — ACCESS CONTROL IN with REPORT ACCESS CONTROLS LOG parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Parameter list header | | | | | | | |
| 0 | (MSB) | ~~additional~~ LOG LIST LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | Reserved | | | | | | | |
| 5 | Reserved | | | | | | LOG PORTION | |
| 6 | (MSB) | COUNTER | | | | | | |
| 7 | | | | | | | | (LSB) |
| | Access Controls Log pages | | | | | | | |
| 8 | Access Controls Log page 0 | | | | | | | |
| | ⋮ | | | | | | | |
| n | Access Controls Log page x | | | | | | | |

The LOG LIST LENGTH field shall contain a count of the number of bytes in the remaining parameter data. The value in the LOG LIST LENGTH field shall be the actual number of bytes available without consideration for insufficient allocation length in the CDB. If access controls are disabled, the LOG LIST LENGTH field shall be set to eight.

The LOG PORTION field (see table t25) indicates the access controls log portion being returned, the contents of the COUNTER field, and the type of Access Controls Log pages being returned.

**Table t25 — Parameter data LOG PORTION field values**

| Log Portion | Access Controls Log Portion Being Returned | COUNTER Field Contents | Access Controls Log Page Format Reference |
|---|---|---|---|
| 00h | Key Overrides portion | Key Overrides counter | 7.1.4.2.2 |
| 01h | Invalid Keys portion | Invalid Keys counter | 7.1.4.2.3 |
| 02h | ACL LUN Conflicts portion | ACL LUN Conflicts counter | 7.1.4.2.4 |
| 11b | Reserved | | |

The COUNTER field contains the events counter value (see 5.99.9) for the access controls log portion indicated by the LOG PORTION field (see table t25).

The format of the Access Controls Log pages is indicated by the value in the LOG PORTION field (see table t25). All the Access Controls Log pages returned in a single parameter list shall have the same format. ~~For any value of the LOG PORTION field, if~~ If the access controls coordinator does not support Access Controls Log pages in the portion of the access controls log indicated by the LOG PORTION field, ~~only supports the relevant event counter in the log and not the additional information, then~~ the ~~returned~~ parameter data shall only contain the parameter list header ~~information~~.

### ~~5.2.4.2 REPORT ACCESS CONTROLS LOG parameter data format for Key Overrides~~

### 7.1.4.2.2 Key Overrides Access Controls Log page format

The Key Overrides Access Controls Log page (see table t26) contains details of recently recorded attempts to override the management identifier key (see 5.99.9) using the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action (see 7.2.8), whether those attempts were successful or not.

**Table t26 — Key Overrides Access Controls Log page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 2 | | | | | | | | |
| 3 | Reserved | | | | | | | SUCCESS |
| 4 | (MSB) | TIME STAMP | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | TRANSPORTID | | | | | | |
| 31 | | | | | | | | (LSB) |
| 32 | (MSB) | INITIAL OVERRIDE LOCKOUT TIMER | | | | | | |
| 33 | | | | | | | | (LSB) |
| 34 | (MSB) | OVERRIDE LOCKOUT TIMER | | | | | | |
| 35 | | | | | | | | (LSB) |

~~The format of the parameter data returned in response to an ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action and LOG PORTION field in the CDB indicating key overrides is shown in Table 18.~~

~~Table 19: REPORT ACCESS CONTROLS LOG parameter data format for key overrides~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~MSB~~ | ~~additional length (*n*-3)~~ | | | | | | |
| ~~3~~ | | | | | | | | ~~LSB~~ |
| ~~4~~ | ~~Reserved~~ | | | | | | | |
| ~~5~~ | ~~Reserved~~ | | | | | | ~~log portion~~ | |
| ~~6~~ | ~~MSB~~ | ~~key overrides counter~~ | | | | | | |
| ~~7~~ | | | | | | | | ~~LSB~~ |
| ~~8~~<br>~~n~~ | ~~Key Overrides Log Page(s)~~ | | | | | | | |

~~The ADDITIONAL LENGTH field shall contain a count of the number of bytes in the remaining parameter data. The value in this field shall contain the actual number of bytes available without consideration for insufficient allocation length.~~

~~The LOG PORTION field shall be set to 00b to indicate which portion of the access controls log is reflected in the rest of the parameter data.~~

~~The KEY OVERRIDES COUNTER field shall contain the Key Overrides Counter maintained by the access controls coordinator.~~

~~The Key Overrides Log Page(s) shall contain a description of the key overrides log entries as recorded by the access controls coordinator (see a.b.c). The format for these pages is found in Table 19.~~

~~Table 20: Key Overrides Log Page(s) data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~2~~ | ~~Reserved~~ | | | | | | | |
| ~~3~~ | ~~Reserved~~ | | | | | | | ~~success~~ |
| ~~4~~<br>~~7~~ | ~~MSB~~ | ~~time stamp~~ | | | | | | ~~LSB~~ |
| ~~8~~<br>~~31~~ | ~~transportid~~ | | | | | | | |
| ~~32~~<br>~~33~~ | ~~MSB~~ | ~~initial override lockout timer~~ | | | | | | ~~LSB~~ |
| ~~34~~<br>~~35~~ | ~~MSB~~ | ~~override lockout timer~~ | | | | | | ~~LSB~~ |

A SUCCESS bit of one indicates that the specific ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action event recorded in the access controls log successfully overrode the management identifier key ~~was successful~~. A value of zero indicates that the command did not succeed.

The TIME STAMP field shall contain zero or an indication of the time at which the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action was processed as described in 5.99.9.

The TRANSPORTID field shall contain the TransportID of the initiator that issued the command.

The INITIAL OVERRIDE LOCKOUT TIMER field shall contain the access controls coordinator's initial override lockout timer value (see 5.99.7.2.2) at time at which the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action was processed. ~~, the OVERRIDE LOCKOUT TIMER field and the TIME STAMP field shall be set to the values for the Initial Override Lockout Timer, Override Lockout Timer and optional time stamp, respectively, at the time the key override event was recorded. See a.b.c.~~

The OVERRIDE LOCKOUT TIMER field shall contain the access controls coordinator's override lockout timer value (see 5.99.7.2.2) at time at which the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action was processed.

### ~~5.2.4.3 REPORT ACCESS CONTROLS LOG parameter data format for Invalid Keys~~

### 7.1.4.2.3 Invalid Keys Access Controls Log page format

The Invalid Keys Access Controls Log page (see table t27) contains details of recently recorded receipts of ACCESS CONTROL IN or ACCESS CONTROL OUT commands specifying an incorrect management identifier key (see 5.99.9).

**Table t27 — Invalid Keys Access Controls Log page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | OPERATION CODE | | | | | | | |
| 3 | Reserved | | | | SERVICE ACTION | | | |
| 4 | (MSB) | TIME STAMP | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | TRANSPORTID | | | | | | |
| 31 | | | | | | | | (LSB) |
| 32 | (MSB) | INVALID MANAGEMENT IDENTIFIER KEY | | | | | | |
| 39 | | | | | | | | (LSB) |

~~The format of the parameter data returned in response to an ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action and LOG PORTION field in the CDB indicating invalid key events is shown in Table 20.~~

~~Table 21: REPORT ACCESS CONTROLS LOG parameter data format for invalid keys~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~MSB~~ | ~~additional length (*n*-3)~~ | | | | | | |
| ~~3~~ | | | | | | | | ~~LSB~~ |
| ~~4~~ | ~~Reserved~~ | | | | | | | |
| ~~5~~ | ~~Reserved~~ | | | | | | ~~log portion~~ | |
| ~~6~~ | ~~MSB~~ | ~~invalid keys counter~~ | | | | | | |
| ~~7~~ | | | | | | | | ~~LSB~~ |
| ~~12~~<br>~~n~~ | ~~Invalid Keys Log Page(s)~~ | | | | | | | |

~~The ADDITIONAL LENGTH field shall contain a count of the number of bytes in the remaining parameter data. The value in this field shall contain the actual number of bytes available without consideration for insufficient allocation length. If access controls are disabled, this field shall be set to zero and at most four (4) bytes of data shall be returned.~~

~~The LOG PORTION field shall be set to 01b to indicate which portion of the access controls log is reflected in the rest of the parameter data.~~

~~The INVALID KEYS COUNTER field shall contain the Invalid Keys Counter maintained by the access controls coordinator.~~

~~The Invalid Keys Log Page(s) shall contain a description of the invalid keys log entries as recorded by the access controls coordinator (see a.b.c). The format for these entries is found in Table 21.~~

~~Table 22: Invalid Keys Log Page(s) data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~ | ~~opcode~~ | | | | | | | |
| ~~3~~ | ~~Reserved~~ | | | ~~service action~~ | | | | |
| ~~4~~<br>~~7~~ | ~~MSB~~ | | | ~~time stamp~~ | | | | ~~LSB~~ |
| ~~8~~<br>~~31~~ | ~~transportid~~ | | | | | | | |
| ~~32~~<br>~~39~~ | ~~MSB~~ | | | ~~invalid key~~ | | | | ~~LSB~~ |

The OPERATION CODE and SERVICE ACTION fields shall be set to the respective values from the CDB of the access controls command that specified ~~contained~~ the invalid management identifier key ~~(in either the CDB or the associated parameter data)~~ whose value in found in the INVALID MANAGEMENT IDENTIFIER KEY field.

~~The TIME STAMP field may be set to the value of the time stamp at the time the invalid key event was recorded. See a.b.c.~~

The TIME STAMP field shall contain zero or an indication of the time at which the ACCESS CONTROL IN or ACCESS CONTROL OUT command was processed as described in 5.99.9.

The TRANSPORTID field shall contain the TransportID of the initiator that issued the command.

The INVALID MANAGEMENT IDENTIFIER KEY field shall be set to the value of the invalid management identifier key detected by the access controls coordinator in the command or associated parameter data.

NOTE 6 - The management identifier key is typically in the CDB for ACCESS CONTROL IN commands and in the parameter data for ACCESS CONTROL OUT commands.

### ~~5.2.4.4 REPORT ACCESS CONTROLS LOG parameter data format for ACL LUN Conflicts~~

### 7.1.4.2.4 ACL LUN Conflicts Access Controls Log page format

The ACL LUN Conflicts Access Controls Log page (see table t28) contains details of recently recorded ACL LUN (see 5.99.9) encountered by the access controls coordinator when a previously not-enrolled initiator sends an ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 7.2.4).

**Table t28 — ACL LUN Conflicts Access Controls Log page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | | | TIME STAMP | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | | | TRANSPORTID | | | | |
| 31 | | | | | | | | (LSB) |
| 32 | (MSB) | | | ACCESSID | | | | |
| 55 | | | | | | | | (LSB) |

~~The format of the parameter data returned in response to an ACCESS CONTROL IN command with REPORT ACCESS CONTROLS LOG service action and LOG PORTION field indicating ACL LUN conflicts is shown in Table 22.~~

~~Table 23: REPORT ACCESS CONTROLS LOG parameter data format for ACL LUN conflicts~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~MSB~~ | | | ~~additional length (*n*-3)~~ | | | | |
| ~~3~~ | | | | | | | | ~~LSB~~ |
| ~~4~~ | ~~Reserved~~ | | | | | | | |
| ~~5~~ | ~~Reserved~~ | | | | | | ~~log portion~~ | |
| ~~6~~ | ~~MSB~~ | | | ~~acl lun conflicts counter~~ | | | | |
| ~~7~~ | | | | | | | | ~~LSB~~ |
| ~~8~~<br>~~n~~ | ~~ACL LUN Conflicts Log Page(s)~~ | | | | | | | |

~~The ADDITIONAL LENGTH field shall contain a count of the number of bytes in the remaining parameter data. The value in this field shall contain the actual number of bytes available without consideration for insufficient allocation length. If access controls are disabled, this field shall be set to zero and at most four (4) bytes of data shall be returned.~~

~~The LOG PORTION field shall be set to 10b to indicate which portion of the access controls log is reflected in the rest of the parameter data.~~

~~The ACL LUN CONFLICTS COUNTER field shall contain the ACL LUN Conflicts Counter maintained by the access controls coordinator.~~

~~The ACL LUN Conflicts Log Page(s) shall contain a description of the ACL LUN conflict log entries as recorded by the access controls coordinator (see a.b.c). The format for these entries is found in Table 23.~~

~~Table 24: ACL LUN Conflicts Log Page(s) data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~3~~ | ~~Reserved~~ | | | | | | | |
| ~~4~~<br>~~7~~ | ~~MSB~~ | | | ~~time stamp~~ | | | | ~~LSB~~ |
| ~~8~~<br>~~31~~ | ~~transportid~~ | | | | | | | |
| ~~32~~<br>~~55~~ | ~~MSB~~ | | | ~~accessid~~ | | | | ~~LSB~~ |

~~The TIME STAMP field may be set to the value of the time stamp at the time the ACL LUN conflict event was recorded. See a.b.c.~~

The TIME STAMP field shall contain zero or an indication of the time at which the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action was processed as described in 5.99.9.

The TRANSPORTID field shall contain the TransportID of the initiator that issued the command.

~~The TRANSPORTID field of the page shall indicate the access identifier (as extracted from the ACL entry) that identifies the initiator that issued the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action that precipitated the ACL LUN conflict event.~~

Editors Note 2 - ROW: I believe that the preceding paragraph is incorrect because some ACL LUN conflict events occur as the result of a proxy LUN. Therefore, making reference to an ACL entry (presumably an ACE) is incorrect because some ACL LUN conflict events do not involve an ACE. I have used the wording from other Access Control Log page formats because it seems to correctly cover all cases.

The ACCESSID field ~~of the page~~ shall be set to the AccessID that the ~~indicated~~ initiator attempted to enroll. This shall correspond to an access identifier in ACL entry at the time the ACL LUN conflict event occurred.

### 7.1.5 REPORT OVERRIDE LOCKOUT TIMER service action ~~(Mandatory)~~

The ACCESS CONTROL IN command with REPORT OVERRIDE LOCKOUT TIMER service action (see table t29) ~~of the ACCESS CONTROL IN command~~ is used ~~by an application client~~ query the state of the override lockout timer (see 5.99.7.2.2). If the ACCESS CONTROL IN command is implemented, the REPORT OVERRIDE LOCKOUT TIMER service action shall be implemented.

**Table t29 — ACCESS CONTROL IN command with REPORT OVERRIDE LOCKOUT TIMER service action**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | OPERATION CODE (86h) | | | | |
| 1 | | Reserved | | | | SERVICE ACTION (03h) | | |
| 2 | (MSB) | | | MANAGEMENT IDENTIFIER KEY | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | (MSB) | | | ALLOCATION LENGTH | | | | |
| 13 | | | | | | | | (LSB) |
| 14 | | | | Reserved | | | | |
| 15 | | | | CONTROL | | | | |

~~The REPORT OVERRIDE LOCKOUT TIMER service action of the ACCESS CONTROL IN command is used by an application client to report on the state of the Override Lockout Timer (see a.b.c and a.b.c).~~

If access controls are disabled, the command shall be terminated with a CHECK CONDITION status, setting the sense key to ILLEGAL REQUEST ~~device server shall ignore the MANAGEMENT IDENTIFIER KEY field and shall respond with GOOD status and returning only subject to the ALLOCATION LENGTH limitation, regardless of the value of any other field in the CDB.~~

~~If access controls are disabled, the device server shall respond with GOOD status and return no data, regardless of the value of any other field in the CDB.~~

Editors Note 3 - ROW: My understanding of the principles of SCSI suggests that returning GOOD status on an ACCESS CONTROLS IN command without returning any parameter data violates the meaning of GOOD status.

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not match~~, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain~~ the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, parameter data shall not be returned, the command shall be terminated~~. If this is not the case, the device server shall return no data and respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense ~~data~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9).

The ALLOCATION LENGTH field is described in 4.3.4.6. The ALLOCATION LENGTH field value ~~in the CDB~~ should be at least eight~~ (8), sufficient for the header information~~.

~~If access controls are enabled and the SERVICE ACTION-SPECIFIC DATA field in the CDB matches the current Management Identifier Key maintained by the access controls coordinator, then the device server shall respond with GOOD status and return the parameter data as specified in Table 24.~~

~~Table 25: MANAGE OVERRIDE LOCKOUT TIMER parameter data format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~MSB~~ | ~~current override lockout timer~~ | | | | | | ~~LSB~~ |
| ~~4~~<br>~~5~~ | ~~MSB~~ | ~~initial override lockout timer~~ | | | | | | ~~LSB~~ |
| ~~6~~<br>~~7~~ | ~~MSB~~ | ~~key overrides counter~~ | | | | | | ~~LSB~~ |

If access controls are enabled, the parameter data returned by the ACCESS CONTROL IN command with REPORT OVERRIDE LOCKOUT TIMER service action shall have the format shown in table t30.

**Table t30 — ACCESS CONTROL IN with REPORT OVERRIDE LOCKOUT TIMER parameter data**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 1 | | | | | | | | |
| 2 | (MSB) | CURRENT OVERRIDE LOCKOUT TIMER | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | INITIAL OVERRIDE LOCKOUT TIMER | | | | | | |
| 5 | | | | | | | | (LSB) |
| 6 | (MSB) | KEY OVERRIDES COUNTER | | | | | | |
| 7 | | | | | | | | (LSB) |

The CURRENT OVERRIDE LOCKOUT TIMER field shall be set to the current value of the override lockout timer (see 5.99.7.2.2).

The INITIAL OVERRIDE LOCKOUT TIMER field shall be set to the value of the initial override lockout timer (see 5.99.7.2.2) as established by the last successful ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action (see 7.2.7).

The KEY OVERRIDES COUNTER field shall be set to the value of the key overrides counter in the access controls log (see 5.99.9).

### 7.1.6 REQUEST PROXY TOKEN service action ~~(Optional)~~

The ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action (see table t31) ~~of the ACCESS CONTROL IN command~~ is used ~~by an initiator~~ to obtain ~~from the access controls coordinator~~ a proxy token (see 5.99.5.2) for a logical unit to which ~~it~~ that initiator has non-proxy access rights. ~~It may use this~~ The proxy token thus obtained may be used to pass temporary access to the logical unit to a third party ~~to grant a third-party temporary access to a logical unit.~~ via ~~This is used in conjunction with the~~ other proxy related service actions of the ACCESS CONTROL IN and ACCESS CONTROL OUT commands. If ~~this~~ the ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action is not supported ~~by the access controls coordinator~~, the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, ~~with~~ the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

**Table t31 — ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (86h) | | | | | | | |
| 1 | Reserved | | | SERVICE ACTION (04h) | | | | |
| 2 | (MSB) | LUN VALUE | | | | | | |
| 9 | | | | | | | | (LSB) |
| 10 | (MSB) | ALLOCATION LENGTH | | | | | | |
| 13 | | | | | | | | (LSB) |
| 14 | Reserved | | | | | | | |
| 15 | CONTROL | | | | | | | |

If access controls are disabled, the command shall be terminated with a CHECK CONDITION status, setting the sense key to ILLEGAL REQUEST ~~device server shall ignore the MANAGEMENT IDENTIFIER KEY field and shall respond with GOOD status and returning only subject to the ALLOCATION LENGTH limitation, regardless of the value of any other field in the CDB.~~

~~If access controls are disabled, the device server shall respond with GOOD status and return no data, regardless of the value of any other field in the CDB.~~

Editors Note 4 - ROW: My understanding of the principles of SCSI suggests that returning GOOD status on an ACCESS CONTROLS IN command without returning any parameter data violates the meaning of GOOD status.

NOTE 7 - ~~The indicated response when access controls are disabled is sufficient for the initiator to determine that access controls are disabled. In this state~~ If access controls are disabled, all logical units are accessible and all initiators share the same LUN values for addressing ~~(this LUN value is the default LUN value)~~. A proxy token is not needed because sharing LUN values is sufficient.~~Consequently, the initiator may use the LUN value to identify the logical unit to a third-party and does not need a Proxy Token.~~

The ~~SERVICE ACTION-SPECIFIC DATA~~ LUN VALUE field shall contain the ~~Logical Unit Number~~ LUN value the initiator uses to access the logical unit for which the proxy token is requested. ~~This LUN should reference the logical unit for which the initiator is requesting the Proxy Token.~~

If the ~~Logical Unit Number~~ LUN value corresponds to a logical unit that is accessible to the requesting initiator either through a TransportID or~~, if the initiator is in the enrolled state,~~ through the AccessID under which ~~it has the~~

initiator is currently in the enrolled state (see 5.99.4.1), and the access controls coordinator has sufficient resources to create and manage a new proxy token, then the parameter data shown in table t32 shall be returned ~~device server shall respond with GOOD status and return in the parameter data an eight (8) byte Proxy Token~~. ~~This token (while valid, see 0.0.19) may be used by a third-party initiator to gain temporary access to the associated logical unit via an ASSIGN PROXY LUN service action.~~

If the ~~Logical Unit Number~~ LUN value does not correspond to an accessible logical unit ~~as indicated above~~, ~~then the following rules apply~~ parameter data shall not be returned and the command shall be terminated as follows:

~~a) If the Logical Unit Number does not correspond to an accessible logical unit, then the device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to ACCESS DENIED - INVALID LU IDENTIFIER;~~

a) If the ~~Logical Unit Number~~ LUN value:
   A) Does not correspond to an accessible logical unit; or
   B) Corresponds to a logical unit accessible only through a proxy token;
   Then the command shall be terminated ~~then the device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INVALID LU IDENTIFIER; or
b) If the ~~Logical Unit Number~~ LUN value corresponds to a logical unit accessible only through an enrolled AccessID ~~for that initiator~~ and the initiator is in the pending-enrolled state, then the command shall be terminated ~~the device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INITIATOR PENDING-ENROLLED.

~~In these cases, no parameter data is returned.~~

If the access controls coordinator does not have enough resources to create and manage a new proxy token, the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INSUFFICIENT ACCESS CONTROL RESOURCES.

The ALLOCATION LENGTH field is described in 4.3.4.6. The ALLOCATION LENGTH field value ~~in the CDB~~ should be at least eight ~~(8), sufficient for a valid Proxy Token~~.

The format of the parameter data returned by the ACCESS CONTROL IN command with REQUEST PROXY TOKEN service action is shown in table t32.

**Table t32 — ACCESS CONTROL IN with REQUEST PROXY TOKEN parameter data**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | | | PROXY TOKEN | | | | |
| 7 | | | | | | | | (LSB) |

## 7.2 ACCESS CONTROL OUT Command

### 7.2.1 ACCESS CONTROL OUT introduction ~~command descriptor block~~

The service actions of the ACCESS CONTROL OUT command (see Table 25) ~~is~~ are used to request service actions by the access controls coordinator to limit or grant access to the logical units to initiators. ~~The command shall be used in conjunction with~~ If the ACCESS CONTROL OUT command is implemented, the ACCESS CONTROL IN command also shall be implemented. ~~This~~ The ACCESS CONTROL OUT command shall not be affected ~~by reservations, persistent reservations or~~ access controls.

**Table t33 — ACCESS CONTROL OUT service actions**

| Service Action | Command name | Type | Reference |
|---|---|---|---|
| 00h | MANAGE ACL | m | 7.2.2 |
| 01h | DISABLE ACCESS CONTROLS | m | 7.2.3 |
| 02h | ACCESS ID ENROLL | m | 7.2.4 |
| 03h | CANCEL ENROLLMENT | m | 7.2.5 |
| 04h | CLEAR ACCESS CONTROLS LOG | m | 7.2.6 |
| 05h | MANAGE OVERRIDE LOCKOUT TIMER | m | 7.2.7 |
| 06h | OVERRIDE MGMT ID KEY | m | 7.2.8 |
| 07h | REVOKE PROXY TOKEN | o | 7.2.9 |
| 08h | REVOKE ALL PROXY TOKENS | o | 7.2.10 |
| 09h | ASSIGN PROXY LUN | o | 7.2.11 |
| 0Ah | RELEASE PROXY LUN | o | 7.2.12 |
| 0Bh - 17h | Reserved | | |
| 18h - 1Fh | Vendor specific | | |

Key: m = Service action implementation is mandatory if ACCESS CONTROL OUT is implemented.
o = Service action implementation is optional.

The CDB format used by all ACCESS CONTROL OUT service actions is shown in table t34.

**Table t34 — ACCESS CONTROL OUT command format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | OPERATION CODE (87h) | | | | | | | |
| 1 | Reserved | | | SERVICE ACTION (see table t33) | | | | |
| 2 | Reserved | | | | | | | |
| 9 | | | | | | | | |
| 10 | (MSB) | PARAMETER LIST LENGTH | | | | | | |
| 13 | | | | | | | | (LSB) |
| 14 | Reserved | | | | | | | |
| 15 | CONTROL | | | | | | | |

If the device contains an access controls coordinator, ~~this~~ the ACCESS CONTROL OUT command shall be processed by the access controls coordinator if addressed to LUN 0. ~~or~~ The ACCESS CONTROL OUT command also may be addressed to any other LUN value whose standard INQUIRY data (see 7.z.z) has the ACC bit set to one, in which case it~~. In the latter case, the command~~ shall be processed in the same manner as if the command had been addressed to LUN 0. If an ACCESS CONTROL OUT command is received by a device server whose standard INQUIRY data has the ACC bit set to zero, the command ~~It~~ shall be ~~rejected by the device server if addressed to any other LUN~~ 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 ~~OPCODE~~.

~~Table 26: ACCESS CONTROL OUT command~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~operation code (87h)~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | ~~service action~~ | | | | |
| ~~2~~<br>~~9~~ | ~~Reserved~~ | | | | | | | |
| ~~10~~<br>~~13~~ | ~~MSB~~ | ~~parameter list length~~ | | | | | | ~~LSB~~ |
| ~~14~~ | ~~Reserved~~ | | | | | | | |
| ~~15~~ | ~~control~~ | | | | | | | |

~~Fields in the ACCESS CONTROL OUT parameter list specify the information required to perform a particular access control service action.~~

The PARAMETER LIST LENGTH field indicates the amount of data that the initiator shall send to the access controls coordinator in the Data-Out buffer. The format of the parameter list is specific to each service action.

~~A description of the additional fields in this command and more details on the PARAMETER LIST LENGTH field are found in the subclause for each service action.~~

## ~~6.2 ACCESS CONTROL OUT Service Actions~~

### ~~6.2.1 ACCESS CONTROL OUT Service Action Codes~~

~~Table 26 gives a list of the ACCESS CONTROL OUT command service action codes.~~

~~Table 27: ACCESS CONTROL OUT command service action codes (M=Mandatory, O=Optional, V=Vendor-specific)~~

| ~~Code~~ | ~~Name~~ | ~~Type~~ | ~~KeyRq~~ | ~~Clause~~ |
|---|---|---|---|---|
| ~~00h~~ | ~~MANAGE ACL~~ | ~~M~~ | ~~Y~~ | ~~0.0.27~~ |
| ~~01h~~ | ~~DISABLE ACCESS CONTROLS~~ | ~~M~~ | ~~Y~~ | ~~0.0.28~~ |
| ~~02h~~ | ~~ACCESS ID ENROLL~~ | ~~M~~ | ~~N~~ | ~~0.0.29~~ |
| ~~03h~~ | ~~CANCEL ENROLLMENT~~ | ~~M~~ | ~~N~~ | ~~0.0.30~~ |
| ~~04h~~ | ~~CLEAR ACCESS CONTROLS LOG~~ | ~~M~~ | ~~Y~~ | ~~0.0.31~~ |
| ~~05h~~ | ~~MANAGE OVERRIDE LOCKOUT TIMER~~ | ~~M~~ | ~~Y~~ | ~~0.0.32~~ |
| ~~06h~~ | ~~OVERRIDE MGMT ID KEY~~ | ~~M~~ | ~~N~~ | ~~0.0.33~~ |
| ~~07h~~ | ~~REVOKE PROXY TOKEN~~ | ~~O~~ | ~~N~~ | ~~0.0.34~~ |
| ~~08h~~ | ~~REVOKE ALL PROXY TOKENS~~ | ~~O~~ | ~~N~~ | ~~0.0.35~~ |
| ~~09h~~ | ~~ASSIGN PROXY LUN~~ | ~~O~~ | ~~N~~ | ~~0.0.36~~ |
| ~~0Ah~~ | ~~RELEASE PROXY LUN~~ | ~~O~~ | ~~N~~ | ~~0.0.37~~ |
| ~~0Bh-17h~~ | ~~Reserved~~ | | | |
| ~~18h-1Fh~~ | ~~Vendor-specific~~ | ~~V~~ | | |

~~The KeyRq column indicates whether the Management Identifier Key shall be supplied for successful completion of the service action (with the exception of special cases where no data is transferred). A “Y” indicates that the Management Identifier Key is required. An “N” indicates that the Management Identifier Key is not required.~~

### 7.2.2 MANAGE ACL service action ~~(Mandatory)~~

#### 7.2.2.1 MANAGE ACL introduction ~~command descriptor block~~

The ACCESS CONTROL OUT command with MANAGE ACL service action ~~version of the ACCESS CONTROL OUT command~~ is used ~~by an application client~~ to authorize access or revoke access to a logical unit or logical units by initiators. ~~This~~ The ACCESS CONTROL OUT command with MANAGE ACL service action adds, changes or removes an entry or multiple entries in the access controls coordinator’s ACL (see 5.99.2). ~~This service action is mandatory~~ If the ACCESS CONTROL OUT command is ~~supported~~ implemented, the MANAGE ACL service action shall be implemented.

The format of the CDB for the ACCESS CONTROL OUT command with MANAGE ACL service action is shown in table t34 (see 7.2.1).

Editors Note 5 - ROW: The way the following paragraph is written (before editing), the ACCESS CONTROL OUT command with MANAGE ACL service action cannot be used enable access controls.

If ~~access controls are disabled or if~~ the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is less than ~~twenty (~~20~~)~~ or results in truncation of any ACE ~~ACL Entry~~ page (see table t36) ~~as specified in 6.2.2.2~~, then the ~~device server shall respond~~ 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 PARAMETER LIST LENGTH ERROR. ~~Otherwise, the structure of the parameter data shall be as described in 0.0.27.2.~~

If the access controls coordinator cannot complete the ACCESS CONTROL OUT command with MANAGE ACL service action because it has insufficient resources~~ to process the command~~, the access controls coordinator shall take no action and not change any of its state and the ~~device server shall return~~ command shall be terminates with a CHECK CONDITION status, the ~~with~~ sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ~~data of~~ INSUFFICIENT ACCESS CONTROL RESOURCES.~~ In this case, no changes shall be made to the access controls coordinator's state.~~

### ~~6.2.2.2 MANAGE ACL parameter list format~~

#### ~~6.2.2.2.1 MANAGE ACL parameter list header~~

~~The format of the parameter list provided for an ACCESS CONTROL OUT command with MANAGE ACL service action is shown in Table 25. The ACL Entry Page(s) are described in 0.0.27.2.2 and 0.0.27.2.3.~~

~~Table 28: MANAGE ACL parameter list format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~3~~ | ~~Reserved~~ | | | | | | | |
| ~~4~~<br>~~11~~ | ~~MSB~~ | ~~management identifier key~~ | | | | | | ~~LSB~~ |
| ~~12~~<br>~~19~~ | ~~MSB~~ | ~~new management identifier key~~ | | | | | | ~~LSB~~ |
| ~~20~~ | ~~Reserved~~ | | | | | | | |
| ~~21~~ | ~~Flush~~ | ~~Reserved~~ | | | | | | |
| ~~22~~<br>~~23~~ | ~~Reserved~~ | | | | | | | |
| ~~24~~<br>~~27~~ | ~~MSB~~ | ~~luns generation~~ | | | | | | ~~LSB~~ |
| ~~28~~<br>~~n~~ | ~~ACL Entry Pages(s)~~ | | | | | | | |

The format of the parameter data for the ACCESS CONTROL OUT command with MANAGE ACL service action is shown in table t35.

**Table t35 — ACCESS CONTROL OUT with MANAGE ACL parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | Parameter list header | | | | | | | |
| 0 | Reserved | | | | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | (MSB) | NEW MANAGEMENT IDENTIFIER KEY | | | | | | |
| 19 | | | | | | | | (LSB) |
| 20 | Reserved | | | | | | | |
| 21 | FLUSH | Reserved | | | | | | |
| 22 | Reserved | | | | | | | |
| 23 | Reserved | | | | | | | |
| 24 | (MSB) | ~~LUNS~~ DLGENERATION | | | | | | |
| 27 | | | | | | | | (LSB) |
| | ~~ACL Entry~~ ACE pages | | | | | | | |
| 28 | ACE page 0 | | | | | | | |
| | | | | | | | | |
| | ⋮ | | | | | | | |
| | ACE page x | | | | | | | |
| n | | | | | | | | |

~~Any of the following conditions in the parameter header or any parameter page require the device server to respond with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST and also make no changes to the access controls coordinator's state:~~

~~a) the INITIATOR TYPE field indicates an unsupported value;~~
~~b) the INITIATOR TYPE=01h (TransportID) and the ACCESS IDENTIFIER field is invalid as specified in the relevant protocol standard;~~
~~c) two ACL Entry Pages contain the same INITIATOR TYPE and ACCESS IDENTIFIER fields;~~
~~d) the LUNS GENERATION field in the header of the parameter data does not match the current value maintained by the access controls coordinator.~~

~~NOTE It is the responsibility of the application client to get (via the REPORT LU DESCRIPTORS service action) the current association of default LUN values to logical units (and the generation value for that association) prior to issuing this service action.~~

~~If the access controls coordinator cannot complete the command because it has insufficient resources to process the command, the device server shall return a CHECK CONDITION with sense key ILLEGAL REQUEST and~~

~~additional sense data of INSUFFICIENT ACCESS CONTROL RESOURCES. In this case, no changes shall be made to the access controls coordinator's state.~~

~~The MANAGEMENT IDENTIFIER KEY field is used to compare with the current Management Identifier Key maintained by the access controls coordinator. If access controls are disabled, then this field is ignored. If access controls are enabled and if the MANAGEMENT IDENTIFIER KEY field in the parameter list does not match the access controls coordinator's current Management Identifier Key, the device server shall return CHECK CONDITION with sense key ILLEGAL REQUEST, additional sense code set to ACCESS DENIED - INVALID MGMT ID KEY and the access controls coordinator shall record the event in the invalid keys portion of the access controls log (see a.b.c) and take no other action. If the access controls coordinator successfully processes the requested service action, the access controls coordinator shall reset its Management Identifier Key to the value specified in the NEW MANAGEMENT IDENTIFIER KEY field and enable access controls.~~

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not match the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, the access controls coordinator's state shall not be altered, the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9).

If the contents of the MANAGEMENT IDENTIFIER KEY field match the current management identifier key maintained by the access controls coordinator, the access controls coordinator shall ~~re~~set its management identifier key to the value specified in the NEW MANAGEMENT IDENTIFIER KEY field and if access controls are disabled it shall enable them ~~access controls~~.

The FLUSH bit of one instructs the access controls coordinator to ~~transition~~ place every initiator in the enrolled state into the pending-enrolled state (see 5.99.4.1.4).

~~The LUNS GENERATION field shall be set to the current value of the Default LUNs Generation integer maintained by the access controls coordinator.~~

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

The ~~ACL Entry page(s)~~ ACE pages that may follow in the parameter list provide additional changes to the ACL. Each ACE page describes one ACE in the ACL that is to be added, modified, or removed. The content and format of an ACE page is indicated by a page code (see table t36).

**Table t36 — ACE page codes**

| Page Code | Description | Reference |
|---|---|---|
| 00h | Grant/Revoke | 7.2.2.2 |
| 01h | Grant All | 7.2.2.3 |
| 02h | Revoke Proxy Token | 7.2.2.4 |
| 03h | Revoke All Proxy Tokens | 7.2.2.5 |
| 04h-EFh | Reserved | |
| F0h-FFh | Vendor-specific | |

Editors Note 6 - ROW: MANAGE ACL Grant/Revoke ACE pages can cause ACL LUN conflicts (see 5.99.4.2) by adding LUN values that are already in use by some initiators via other ACEs or proxy LUNs. This condition is not covered in 5.99.4.2 or here. My understanding is that the authors' intention was for such ACL LUN conflicts to result in termination of the ACCESS CONTROL OUT command with

MANAGE ACL service action.

The following requirements apply to the processing of changes to the access control state ~~of the device follow these rules~~:

a) No change to the access control state ~~of the device~~ shall occur if the ACCESS CONTROL OUT command with MANAGE ACL service action terminates with a status other than ~~cannot be processed with~~ GOOD status; and
b) If the ACCESS CONTROL OUT command with MANAGE ACL service action completes with a ~~results in~~ GOOD status, the following shall have been performed ~~be instantiated~~ as a single indivisible event:
   1) Changes resulting from the contents of ~~dictated in the~~ fields in the parameter list header shall be ~~of the parameter list are~~ processed; and
   2) Changes resulting from the contents of ~~dictated by~~ ~~ACL Entry~~ ACE pages ~~are~~ shall be processed;
      a) Multiple ~~ACL Entry~~ ACE pages ~~are~~ shall be processed sequentially;
      b) If an ~~ACL Entry~~ ACE page contains conflicting instructions in LUACD descriptors, the ~~last~~ instructions in the last LUACD descriptor within the page shall take~~s~~ precedence; and
      c) If an ACE containing an AccessID type access identifier (see 5.99.2.2.2) ~~AccessID's ACL entry~~ is replaced and the ACE page that caused the change has the NOCNCL bit (see 7.2.2.2) set to zero ~~(see 6.2.2.2.2)~~, then ~~the~~ any initiator in the enrolled or pending-enrolled state under ~~that~~ the AccessID in that ACE shall be ~~transitioned to~~ placed in the not-enrolled state (see 5.99.4.1.2)~~, unless indicated otherwise by a NOCNCL bit value of one in the ACL Entry Page (see 0.0.27.2.2)~~.

An ~~ACL Entry~~ ACE page contains conflicting instructions if either of the following ~~occurs~~ is true:

a) Two LUACD descriptors ~~LUN/default LUN pairs appear~~ are present with the same LUN value and different default LUN values; or
b) Two LUACD descriptors ~~LUN/default LUN pairs appear~~ are present with different LUN values and the same default LUN value.

~~The structure of ACL Entry pages and the action to be taken is determined by a PAGE CODE field as defined in Table 28. Details of the contents of each page are described in subsequent subclauses.~~

~~Table 29: ACL Entry Page Code definitions~~

| ~~Page Code~~ | ~~Action~~ | ~~Clause~~ |
|---|---|---|
| ~~00h~~ | ~~Grant/Revoke~~ | ~~0.0.27.2.2~~ |
| ~~01h~~ | ~~Grant All~~ | ~~0.0.27.2.2~~ |
| ~~02h~~ | ~~Revoke Proxy Token~~ | ~~0.0.27.2.3~~ |
| ~~03h~~ | ~~Revoke All Proxy Tokens~~ | ~~0.0.27.2.3~~ |
| ~~04h-EFh~~ | ~~Reserved~~ | |
| ~~F0h-FFh~~ | ~~Vendor-specific~~ | |

### ~~6.2.2.2.2 MANAGE ACL parameter data Grant/Revoke and Grant All page formats~~

~~The Grant/Revoke and Grant All page formats for the MANAGE ACL service action is given in Table 27.~~

~~Table 30: Grant/Revoke and Grant All page formats~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~page code (00h-01h)~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~page length (m-3)~~ | | | | | | | |
| ~~4~~ | ~~NoCncl~~ | ~~Reserved~~ | | | | | | |
| ~~5~~ | ~~identifier type~~ | | | | | | | |
| ~~6~~<br>~~7~~ | ~~identifier length (n-7)~~ | | | | | | | |
| ~~8~~<br>~~n~~ | ~~MSB~~<br>~~access identifier~~ | | | | | | | ~~LSB~~ |
| ~~n+1~~<br>~~m~~ | ~~lun/default lun list~~ | | | | | | | |

### 7.2.2.2 The Grant/Revoke ACE page

The Grant/Revoke ACE page (see table t37) is used to add, modify, or remove an ACE from the ACL (see 5.99.2).

**Table t37 — Grant/Revoke ACE page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (00h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | PAGE LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | NOCNCL | Reserved | | | | | | |
| 5 | ACCESS IDENTIFIER TYPE | | | | | | | |
| 6 | (MSB) | ACCESS IDENTIFIER LENGTH (m-7) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | ACCESS IDENTIFIER | | | | | | | |
| m | | | | | | | | |
| | LUACD Descriptors | | | | | | | |
| m+1 | LUACD descriptor 0 | | | | | | | |
| m+20 | | | | | | | | |
| | ⋮ | | | | | | | |
| n-19 | LUACD descriptor x | | | | | | | |
| n | | | | | | | | |

The PAGE LENGTH field specifies ~~shall indicate~~ the number of additional bytes ~~required for~~ present in this page.

A NOCNCL (no changes to current logical unit access) bit of one specifies that the application client believes that this ACE page makes no changes to the existing logical unit access conditions in the ACL. A NOCNCL bit of zero specifies that the ACE page may or may not change existing logical unit access conditions. If the ACCESS IDENTIFIER TYPE specifies ~~indicates type~~ a TransportID (see 7.1.2.2.2), ~~then~~ the NOCNCL bit ~~is~~ shall be ignored.

~~The IDENTIFIER TYPE and ACCESS IDENTIFIER fields are described in a.b.c. The IDENTIFIER LENGTH field indicates the number of bytes following taken up by the ACCESS IDENTIFIER field.~~

The ACCESS IDENTIFIER TYPE and ACCESS IDENTIFIER length fields are described in 7.1.2.2.2.

The ACCESS IDENTIFIER field contains the identifier that the access controls coordinator uses to select the ACE that is to be added, modified, or removed. The format of the ACCESS IDENTIFIER field is specified in table t12 (see 7.1.2.2.2).

Any of the following conditions in the parameter header or any ~~parameter~~ Grant/Revoke ACE page or Grant All ACE page shall cause the access coordinator to not change its state and shall cause the command to be terminated ~~require the device server to respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST ~~and also make no changes to the access controls coordinator's state~~:

a) The contents of the DLGENERATION field in the parameter list header (see 7.2.2.1) do not match the current DLgeneration value (see 5.99.3.4) maintained by the access controls coordinator;
b) An ~~the INITIATOR~~ ACCESS IDENTIFIER TYPE field ~~indicates~~ that specifies an unsupported value;
c) An ~~the INITIATOR~~ ACCESS IDENTIFIER TYPE that contains =01h (see 5.99.2.2) ~~(TransportID) and the~~ with an ACCESS IDENTIFIER field ~~is invalid as specified in the relevant~~ that contains an invalid TransportID (see 5.99.2.2.3) as defined for the applicable protocol standard; or
d) Two ~~ACL Entry~~ ACE pages that have the same values in the ~~contain the same INITIATOR~~ ACCESS IDENTIFIER TYPE and ACCESS IDENTIFIER fields.

NOTE 8 - ~~It is the responsibility of the~~ The application client is responsible for obtaining ~~to get (via the REPORT LU DESCRIPTORS service action)~~ the current association of default LUN values to logical units (and the DLgeneration value for that association) prior to issuing this service action. The ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action (see 7.1.3) returns the necessary information.

Each LUACD descriptor (see table t38) describes the access to be allowed to one logical unit based on the access identifier in the ACE page. An ACE page may contain zero or more LUACD descriptors.

**Table t38 — ACE page LUACD descriptor format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | ACCESS MODE | | | | | | | |
| 1 | Reserved | | | | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | LUN VALUE | | | | | | |
| 11 | | | | | | | | (LSB) |
| 12 | (MSB) | DEFAULT LUN | | | | | | |
| 19 | | | | | | | | (LSB) |

The ACCESS MODE field is described in 7.1.2.2.2.

The LUN VALUE field specifies the LUN value an accessing initiator uses to access the logical unit to which the LUACD descriptor applies.

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 5.99.3.3). The value in the DEFAULT LUN field shall be consistent with the DLGENERATION field contents specified in the parameter list header (see 7.2.2.1).

If the specified access mode ~~value occurs that~~ is not supported or if ~~any~~ the DEFAULT LUN field contains value that is not valid ~~at the access controls coordinator~~ or ~~any~~ the LUN VALUE field contains a value that the access controls coordinator does not support ~~cannot be supported~~ as a valid LUN ~~address~~, the access controls coordinator's state shall not be modified and the command shall be terminated ~~the device server shall fail the command~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, ~~and~~ the additional sense code shall be set to ACCESS DENIED - INVALID LU IDENTIFIER, and the SENSE-KEY SPECIFIC field shall be set as described for the ILLEGAL REQUEST sense key in 7.z.z. ~~The sense data shall be modified as follows. The SENSE-KEY SPECIFIC bit~~

~~shall be set as described in 7.20.2 (of SPC-3, revision 0) with the FIELD POINTER field indicating the first byte of the invalid field (as counted within the full parameter data).~~ If the error is ~~caused by~~ an unsupported value in the LUN VALUE field, the first ~~next~~ eight bytes of the additional sense bytes should contain ~~(if available) beyond the last byte of the FIELD POINTER may include~~ a suggested LUN value that the access controls coordinator ~~would~~ supports ~~for the logical unit referenced by the paired default LUN~~.

Based on the access identifier and the presence or absence of LUACD descriptors, the access controls coordinator shall add, modify, or remove an ACE in the ACL as shown in table t39.

**Table t39 — Access Coordinator Grant/Revoke ACE page actions**

| | | ACL already contains an ACE with the access identifier matching the one in the ACE page? | |
|---|---|---|---|
| | | **Yes** | **No** |
| **ACE page includes LUCAD descriptors?** | **Yes** | Modify the existing ACE in the ACL. | Add a new ACE to the ACL. |
| | **No** | Remove the existing ACE from the ACL. | Take no action, this shall not be considered a error. |

~~NOTE All currently defined Identifier Types require the IDENTIFIER LENGTH field be set to 24 (see Table 35).~~

~~The PAGE LENGTH field shall indicate the number of additional bytes required for this page.~~

~~For the Grant/Revoke page, the LUN/DEFAULT LUN LIST field shall contain a (possibly empty) set of LUN/default LUN pairs as specified in Table 10. If an ACCESS MODE value occurs that is not supported or if any default LUN value is not valid at the access controls coordinator or any LUN value cannot be supported as a valid LUN address, the device server shall fail the command with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INVALID LU IDENTIFIER. The sense data shall be modified as follows. The SENSE-KEY SPECIFIC bit shall be set as described in 7.20.2 (of SPC-3, revision 0) with the FIELD POINTER field indicating the first byte of the invalid field (as counted within the full parameter data). If the error is caused by an unsupported LUN value, the next eight bytes (if available) beyond the last byte of the FIELD POINTER may include a LUN value that the access controls coordinator would support for the logical unit referenced by the paired default LUN.~~

~~A Grant/Revoke page with a non-empty LUN/DEFAULT LUN LIST instructs the access controls coordinator to add a new or to replace an existing ACL entry for the specified access identifier. The accessible logical unit pairs of this ACL entry shall be derived from the LUN/DEFAULT LUN LIST as follows. Each accessible logical unit pair shall take its LUN value from a LUN/default LUN pair and its logical unit reference shall refer to the logical unit corresponding to the default LUN value. The access rights to that logical unit shall be as specified by the Access Mode value as described in 0.0.21.2.2.~~

~~A Grant/Revoke page with an empty LUN/DEFAULT LUN LIST instructs the access controls coordinator to remove an existing ACL entry for the specified access identifier. It is not an error condition if no such entry exists.~~

~~The Grant All page shall contain an empty LUN/DEFAULT LUN LIST field. That is, there shall be no data in this page after the last byte of the ACCESS IDENTIFIER field.~~

~~The Grant All page instructs the access controls coordinator to add a new or replace an existing ACL entry for the specified access identifier. The Grant All page shall be processed to have the same effect as a Grant/Revoke page containing the same access identifier and a complete list of LUN/default LUN pairs (with LUN equal to the default LUN in each pair and with ACCESS MODE set to 00h, normal access) for all logical units.~~

~~NOTE A Grant All page has the effect that any initiator associated with the access identifier shall have the same access to logical units and the same INQUIRY and REPORT LUNS response as if access controls were disabled.~~

~~If the IDENTIFIER TYPE indicates type TransportID, then the NOCNCL bit is ignored.~~

If the ACCESS IDENTIFIER TYPE indicates type AccessID, the enrollment state (see 5.99.4.1) of any ~~and an~~ initiator that is enrolled ~~(in either the enrolled or pending-enrolled state)~~ under the specified AccessID, ~~then the initiator's enrollment state~~ shall be affected ~~according to the following rules (see also 0.0.14.2)~~ as follows:

a) If the ACE containing the ~~ACL entry corresponding to that~~ AccessID is removed ~~as a consequence of the Grant/Revoke page~~, the initiator shall be placed in ~~is transitioned to~~ the not-enrolled state; or

b) If the ACE containing the ~~ACL entry corresponding to that~~ AccessID is modified by a ~~replaced as a consequence of the~~ Grant/Revoke ACE page or a Grant All ACE page, then;

   A) If the NOCNCL bit is zero in that ACE page, the initiator shall be placed in ~~is transitioned to~~ the not-enrolled state; or

   B) If ~~the ACL entry corresponding to that AccessID is replaced as a consequence of the Grant/Revoke or Grant All page and~~ the NOCNCL bit is one in that ACE page, ~~then~~ the enrollment state of the initiator may be left unchanged or the initiator may be placed in ~~transitioned to~~ the not-enrolled state (see 5.99.4.1.2) ~~in a vendor-specific manner~~.

### 7.2.2.3 The Grant All ACE page

The Grant All ACE page (see table t40) is used to add or modify an ACE from the ACL (see 5.99.2). An ACE added or modified using the Grant All ACE page allows initiators with the specified access identifier to access the SCSI target device as if access controls were disabled.

**Table t40 — Grant All ACE page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (01h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | PAGE LENGTH (n-3) | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | NOCNCL | Reserved | | | | | | |
| 5 | ACCESS IDENTIFIER TYPE | | | | | | | |
| 6 | (MSB) | ACCESS IDENTIFIER LENGTH (m-7) | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | ACCESS IDENTIFIER | | | | | | | |
| n | | | | | | | | |

The PAGE LENGTH, ACCESS IDENTIFIER TYPE, ACCESS IDENTIFIER LENGTH, and ACCESS IDENTIFIER fields are defined in 7.2.2.3.

When an existing ACE that was created or modified using the Grant/Revoke ACE page is modified by a Grant All ACE page or when an existing ACE that was created or modified using the Grant All ACE page is modified by a Grant/Revoke ACE page, the modification shall be processed as if the Grant All ACE page is or was a Grant/

Revoke ACE page with one LUACD descriptor for every logical unit managed by the access controls coordinator with the fields in each LUACD containing:

a) An access mode of 00h (see 7.1.2.2.2);
b) A LUN VALUE field whose contents match the contents of the DEFAULT LUN field; and
c) A DEFAULT LUN field whose contents reference the logical unit appropriate to the DLgeneration value (see 5.99.3.3).

~~The Grant All page instructs the access controls coordinator to add a new or replace an existing ACL entry for the specified access identifier. The Grant All page shall be processed to have the same effect as a Grant/Revoke page containing the same access identifier and a complete list of LUN/default LUN pairs (with LUN equal to the default LUN in each pair and with ACCESS MODE set to 00h, normal access) for all logical units.~~

~~NOTE A Grant All page has the effect that any initiator associated with the access identifier shall have the same access to logical units and the same INQUIRY and REPORT LUNS response as if access controls were disabled.~~

### ~~6.2.2.3 MANAGE ACL parameter data Revoke Proxy Token and Revoke All Proxy Tokens page formats~~

~~The Revoke Proxy Token and Revoke All Proxy Tokens page formats for the MANAGE ACL service action is given in Table 28.~~

~~Table 31: Revoke Proxy Token and Revoke All Proxy Tokens page formats~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~page code (02h–03h)~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~page length (m–3)~~ | | | | | | | |
| ~~4~~<br>~~m~~ | ~~proxy token list~~ | | | | | | | |

### 7.2.2.4 The Revoke Proxy Token ACE page

The Revoke Proxy Token ACE page (see table t41) is used to revoke one or more proxy tokens (see 5.99.5.2).

**Table t41 — Revoke Proxy Token ACE page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (02h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | PAGE LENGTH (n-3) | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | PROXY TOKEN 0 | | | | | | | |
| 11 | | | | | | | | |
| | ⋮ | | | | | | | |
| n-7 | PROXY TOKEN x | | | | | | | |
| n | | | | | | | | |

The PAGE LENGTH field specifies ~~shall indicate~~ the number of additional bytes ~~required for~~ present in this page.

~~For the Revoke Proxy Token page, the PROXY TOKEN LIST field shall contain a list of Proxy Tokens (eight (8) bytes each). This instructs the~~ The one or more PROXY TOKEN field(s) specify the proxy tokens to be revoked. The access controls coordinator ~~to~~ shall revoke each proxy token ~~of the~~ listed in a PROXY TOKEN field. ~~It is not an error condition if a Proxy Token specified in this page is not currently valid. In this case, no action is taken by the access controls coordinator with respect to this token.~~ If the contents of a PROXY TOKEN field do not identify a valid proxy token the field shall be ignored, this shall not be considered an error.

~~For the Revoke All Proxy Tokens page, the PROXY TOKEN LIST field shall be empty. This instructs the access controls coordinator to revoke all existing Proxy Tokens.~~

Multiple Revoke Proxy Token ~~and Revoke All Proxy Tokens~~ ACE pages may be included in the parameter data. ~~They are processed sequentially.~~

### 7.2.2.5 The Revoke All Proxy Tokens ACE page

The Revoke All Proxy Tokens ACE page (see table t41) is used to revoke all currently valid proxy tokens (see 5.99.5.2).

**Table t42 — Revoke All Proxy Tokens ACE page format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PAGE CODE (03h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | | | PAGE LENGTH (0000h) | | | | |
| 3 | | | | | | | | (LSB) |

Multiple Revoke ALL Proxy Tokens ACE pages may be included in the parameter data.

### 7.2.3 DISABLE ACCESS CONTROLS service action ~~(Mandatory)~~

The ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an application client~~ to ~~return~~ place the access controls coordinator ~~to~~ in access controls disabled state. If the ACCESS CONTROL OUT command is implemented, the DISABLE ACCESS CONTROLS service action shall be implemented.

The format of the CDB for the ACCESS CONTROL OUT command with DISABLE ACCESS CONTROLS service action is shown in table t34 (see 7.2.1).

If access controls are disabled or if the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor ~~twelve (~~12~~)~~, the ~~device server shall respond~~ 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 PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is ~~twelve (~~12~~)~~, the parameter list shall have the format shown ~~be as described~~ in table t43.

**Table t43 — ACCESS CONTROL OUT with DISABLE ACCESS CONTROLS parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 11 | | | | | | | | (LSB) |

~~Table 32: DISABLE ACCESS CONTROLS parameter list format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~3~~ | ~~Reserved~~ | | | | | | | |
| ~~4~~<br>~~11~~ | ~~MSB~~ | ~~management identifier key~~ | | | | | | ~~LSB~~ |

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not ~~shall~~ match the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, the access controls coordinator's states shall not be altered, the command shall be terminated with a ~~. If this is not the case, the device server shall return~~ CHECK CONDITION status, the ~~with~~ sense key shall be set to ILLEGAL REQUEST, the additional sense ~~data of~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9) ~~and take no other action~~.

~~If access controls are enabled and the Management Identifier Key field matches the current Management Identifier Key maintained by the access controls coordinator, the device server shall respond with GOOD status and the access controls coordinator shall…~~

In response to a 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 5.99.2);
c) Place ~~transition~~ all initiators into the not-enrolled state (see 5.99.4.1);
d) Set the management identifier key to zero (see 5.99.7);
e) Set the override lockout timer to zero (see 5.99.7.2.2);
f) Set the initial override lockout timer value to zero (see 5.99.7.2.2);
g) Clear the access controls log (including resetting counters to zero) with the exception of the key overrides portion of the access controls log (see 5.99.9);
h) Allow all initiator's access to all logical units at their default LUN value; and
i) Optionally, ~~the access controls coordinator may~~ reset the ~~Default LUNs Generation~~ DLgeneration value to zero (see 5.99.3.4).

### 7.2.4 ACCESS ID ENROLL service action ~~(Mandatory)~~

The ACCESS ID ENROLL service action of the ACCESS CONTROL OUT command is used by an initiator to enroll an AccessID with the access controls coordinator. ~~This service action is mandatory~~ If the ACCESS CONTROL OUT command is ~~supported~~ implemented, the ACCESS ID ENROLL service action shall be implemented.

The format of the CDB for the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action is shown in table t34 (see 7.2.1).

If access controls are disabled or if the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor ~~twenty-four (~~24~~)~~, the ~~device server shall respond~~ 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 PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is ~~twenty-four (~~24~~)~~, the parameter list shall ~~contain the AccessID in the format of Table 37.~~ have the format shown in table t44.

**Table t44 — ACCESS CONTROL OUT with ACCESS ID ENROLL parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | | | | ACCESSID | | | | |
| 15 | | | | | | | | |
| 16 | (MSB) | | | Reserved | | | | |
| 23 | | | | | | | | (LSB) |

The AccessID field is described in 5.99.2.2.2.

If the initiator is in the enrolled or pending-enrolled state (see 5.99.4.1) under a given AccessID and the ~~parameter data~~ ACCESSID field contains a different AccessID, the access controls coordinator shall place the initiator in the pending-enrolled state, ~~then the device server shall respond~~ the command shall be terminated with a CHECK CONDITION status, ~~with~~ the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - ENROLLMENT CONFLICT. ~~Additionally, the access controls coordinator shall place the initiator in the pending-enrolled state.~~

If the initiator is in the enrolled or pending-enrolled state under a given AccessID and the ~~parameter data~~ ACCESSID field contains a matching AccessID, ~~then the device server shall respond with GOOD status, and~~ the access controls coordinator shall place the initiator in the enrolled state and make no other changes ~~change to the access rights for that initiator~~.

If the initiator is in the not-enrolled state and the ACCESSID field contents do not match the AccessID in any ACE in the ACL (see 5.99.2) ~~AccessID in the parameter data has no access rights associated with it~~, ~~then~~ the initiator ~~stays~~ shall remain in the not-enrolled state and the ~~device server responds~~ 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 ACCESS DENIED - NO ACCESS RIGHTS.

~~If the following hold:~~

~~a) the initiator is in the not-enrolled state;~~
~~b) the AccessID in the parameter data a corresponding entry in the ACL;~~
~~c) the enrollment does not create a ACL LUN conflict (see 0.0.15),~~

~~then the device server shall respond with GOOD status and the access controls coordinator shall place the initiator into the enrolled state according to the specification in a.b.c.~~

~~If the following hold:~~

~~a) the initiator is in the not-enrolled state;~~
~~b) the AccessID in the parameter data has a corresponding entry in the ACL;~~
~~c) the enrollment creates a ACL LUN conflict (see 0.0.15).~~

~~then the device server shall respond with a CHECK CONDITION status, and sense key set to ILLEGAL REQUEST, with additional sense code set to ACCESS DENIED - ACL LUN CONFLICT and the access controls coordinator shall leave the initiator in the not-enrolled state and record the event in the ACL LUN conflicts portion of the access controls log.~~

If the initiator is in the not-enrolled state and the ACCESSID field contents matches the AccessID in any ACE in the ACL the actions taken depend on whether enrolling the initiator would create an ACL LUN conflict (see 5.99.4.2). If there is no ACL LUN conflict, the initiator shall be placed in the enrolled state (see 5.99.4.1.3). If there is an ACL LUN conflict, the initiator shall remain in the not-enrolled state and the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, the additional sense code shall be set to ACCESS DENIED - ACL LUN CONFLICT and the event shall be recorded in the ACL LUN conflicts portion of the access controls log (see 5.99.9).

> NOTE 9 - ~~If~~ An initiator that receives the ACCESS DENIED - ACL LUN CONFLICT additional sense code ~~data, it~~ should remove any proxy access rights it has acquired using the ACCESS CONTROL OUT command with RELEASE PROXY LUN service action and ~~then~~ retry the enrollment request. If the ACL LUN conflict resulted from proxy access, the retried enrollment succeeds. Otherwise, the mechanisms for resolving ACL LUN conflicts are outside the scope of this standard. ~~(This is recommended in order to verify whether the conflict occurred because of its proxy rights.) If the enrollment fails again, the initiator may (through means beyond the scope of this standard) inform the application client managing access controls that a conflict occurred (because of the state of the ACL) so that the application client may take whatever corrective action is necessary.~~

~~If the AccessID in the parameter data has no access rights associated with it, then the initiator stays in the not-enrolled state and the device server responds with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - NO ACCESS RIGHTS.~~

### 7.2.5 CANCEL ENROLLMENT service action ~~(Mandatory)~~

The ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an initiator~~ to remove ~~its~~ an initiator's enrollment with the access controls coordinator (see 5.99.4). Successful completion of this command changes the state of the initiator to the not-enrolled state. If the ACCESS CONTROL OUT command is implemented, the CANCEL ENROLLMENT service action shall be implemented.

~~This~~ The ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action should be used by an initiator prior to any period where use of its accessible logical units ~~will~~ may be suspended for ~~an extensive~~ a lengthy period of time (e.g., ~~if the~~ when a host is preparing to shutdown). This allows the access controls coordinator to free any resources allocated to manage the enrollment for that initiator.

The format of the CDB for the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action is shown in table t34 (see 7.2.1).

If access controls are disabled, the access controls coordinator shall take no action and the command shall be completed with a GOOD status.

There is no parameter data for ~~this~~ the ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action. If the PARAMETER LIST LENGTH field in the CDB ~~for this service action shall be~~ is not set to zero, the initiator's enrollment shall not be changed and the command shall be terminated ~~. If not, the device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the ~~Parameter List Length field~~ PARAMETER LIST LENGTH field in the CDB is set to zero, the initiator shall be placed in the not-enrolled state (see 5.99.4.1.2) ~~then the device server shall always return GOOD status regardless of the enrolled state of the initiator, unless otherwise specified in this subclause.~~ Any subsequent commands addressed to the logical units no longer accessible are handled according to the rules ~~of~~ stated in 5.99.6.

### 7.2.6 CLEAR ACCESS CONTROLS LOG service action ~~(Mandatory)~~

The ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an application client~~ to instruct the access controls coordinator to reset a specific access control log counter to zero and to clear a portion of the access controls log (see 5.99.9). If the ACCESS CONTROL OUT command is implemented, the CLEAR ACCESS CONTROLS LOG service action shall be implemented.

The format of the CDB for the ACCESS CONTROL OUT command with CLEAR ACCESS CONTROLS LOG service action is shown in table t34 (see 7.2.1).

If access controls are disabled or if the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor ~~twelve (~~12~~)~~, the ~~device server shall respond~~ 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 PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is ~~twelve (~~12~~)~~, the parameter list shall have the format shown ~~be as described~~ in table t45.

**Table t45 — ACCESS CONTROL OUT with CLEAR ACCESS CONTROLS LOG parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 2 | | | | | | | | |
| 3 | Reserved | | | | | | LOG PORTION | |
| 4 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 11 | | | | | | | | (LSB) |

~~Table 33: CLEAR ACCESS CONTROLS LOG parameter list format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~2~~ | ~~Reserved~~ | | | | | | | |
| ~~3~~ | ~~Reserved~~ | | | | | | ~~log portion~~ | |
| ~~4~~<br>~~11~~ | ~~MSB~~ | ~~management identifier key~~ | | | | | | ~~LSB~~ |

~~The LOG PORTION field of this structure shall be interpreted according to Table 17.~~

~~The LOG PORTION field shall not indicate key overrides (00b). If this is the case, then the device server shall return CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.~~

The LOG PORTION field (see table t46) specifies the access controls log portion to be cleared.

**Table t46 — CLEAR ACCESS CONTROLS LOG LOG PORTION field values**

| Log Portion | Description |
|---|---|
| 00b | Reserved |
| 01b | Invalid Keys portion |
| 10b | ACL LUN Conflicts portion |
| 11b | Reserved |

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not ~~shall~~ match the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, the access controls coordinator's states shall not be altered, the command shall be terminated with a ~~. If this is not the case, the device server shall return~~ CHECK CONDITION status, the ~~with~~ sense key shall be set to ILLEGAL REQUEST, the additional sense ~~data of~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9) ~~and take no other action~~.

~~If access controls are enabled and the MANAGEMENT IDENTIFIER KEY value matches the current Management Identifier Key, then the following shall be performed by the access controls coordinator for that portion of the access controls log specified by the LOG PORTION value (when not indicating key overrides):~~

In response to a 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 t46) in the parameter data:

a) Set ~~preset the access controls log~~ the counter for the specified log portion to zero; and
b) If the specified log portion contains details records, remove ~~clear~~ the detail records from the specified log portion ~~additional access controls log information~~.

~~In this case, the device server shall respond with GOOD status.~~

### 7.2.7 MANAGE OVERRIDE LOCKOUT TIMER service action ~~(Mandatory)~~

The ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an application client~~ to manage the override lockout timer (see 5.99.7.2.2). If the ACCESS CONTROL OUT command is implemented, the MANAGE OVERRIDE LOCKOUT TIMER service action shall be implemented.

If access controls are disabled, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status~~, regardless of the value of any other field in the CDB~~.

The format of the CDB for the ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action is shown in table t34 (see 7.2.1).

If the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall ~~restart~~ reset the override lockout timer ~~(reset the value of this timer~~ to the current initial override lockout timer value maintained by the access controls coordinator ~~) and the device server shall respond with GOOD status~~.

If the value in the PARAMETER LIST LENGTH field is neither zero nor ~~twelve (~~12~~)~~, the device server shall respond with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is ~~twelve (~~12~~)~~, the parameter list shall have the format shown ~~be as described~~ in table t47.

**Table t47 — ACCESS CONTROL OUT with MANAGE OVERRIDE LOCKOUT TIMER parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 1 | | | | | | | | |
| 2 | (MSB) | NEW INITIAL OVERRIDE LOCKOUT TIMER | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | MANAGEMENT IDENTIFIER KEY | | | | | | |
| 11 | | | | | | | | (LSB) |

~~Table 34: MANAGE OVERRIDE LOCKOUT TIMER parameter list format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~MSB~~ | ~~new initial override lockout timer~~ | | | | | | ~~LSB~~ |
| ~~4~~<br>~~11~~ | ~~MSB~~ | ~~management identifier key~~ | | | | | | ~~LSB~~ |

The NEW INITIAL OVERRIDE LOCKOUT TIMER field specifies the value that access controls coordinator maintains for initial override lockout timer if the specified management identifier key is correct.

If access controls are enabled and the contents of the MANAGEMENT IDENTIFIER KEY field do not ~~shall~~ match the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator, the access controls coordinator shall not change the initial override lockout timer value but shall set the override lockout timer to the unaltered ~~its~~ current initial override lockout timer value. The command shall be terminated with a ~~. If this is not the case, the device server shall return~~ CHECK CONDITION status, the ~~with~~ sense key shall be set to ILLEGAL REQUEST, the additional sense ~~data of~~ code shall be set to ACCESS DENIED - INVALID MGMT ID KEY, and ~~the access controls coordinator shall record~~ the event shall be recorded in the invalid keys portion of the access controls log (see 5.99.9)~~, reset the Override Lockout Timer to its current Initial Override Lockout Timer value and take no other action~~.

~~If the MANAGEMENT IDENTIFIER KEY value matches the current Management Identifier Key, then the following shall be performed by the access controls coordinator:~~

In response to a ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action with correct management identifier key value the access controls coordinator shall:

a) Replace the currently saved ~~preset the~~ initial override lockout timer with the value in ~~to the value of~~ the NEW INITIAL OVERRIDE LOCKOUT TIMER field ~~in parameter data~~; and
b) Set ~~reset~~ the override lockout timer to the new initial value.

~~In this case, the device server shall respond with GOOD status.~~

### 7.2.8 OVERRIDE MGMT ID KEY service action ~~(Mandatory)~~

The ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an application client~~ to override the current management identifier key (see 5.99.3.2) maintained by the access controls coordinator. ~~This~~ The ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action is intended to be used in a failure situation where the ~~managing~~ application client no longer has access to its copy of this key. Successful use of the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action is restricted by the override lockout timer (see 5.99.7.2.2). If the ACCESS CONTROL OUT command is implemented, the OVERRIDE MGMT ID KEY service action shall be implemented.

The format of the CDB for the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action is shown in table t34 (see 7.2.1).

If access controls are disabled or if the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If access controls are enabled, the access controls coordinator shall log ~~the event~~ every ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action processed whether successful or not in the access controls log as specified in 5.99.9.

~~If access controls are enabled, successful completion of this service action depends on the state of the Override Lockout Timer managed by the access controls coordinator. In any case, the access controls coordinator shall log the event in the access controls log as specified in a.b.c.~~

If the value in the PARAMETER LIST LENGTH field is neither zero nor ~~twelve (~~12~~)~~, the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is ~~twelve (~~12~~)~~, the parameter data shall have the format shown ~~be as described~~ in table t48.

**Table t48 — ACCESS CONTROL OUT with OVERRIDE MGMT ID KEY parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | Reserved | | | | | | | |
| 3 | | | | | | | | |
| 4 | (MSB) | NEW MANAGEMENT IDENTIFIER KEY | | | | | | |
| 11 | | | | | | | | (LSB) |

~~Table 35: OVERRIDE MGMT ID KEY parameter list format~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~3~~ | ~~Reserved~~ | | | | | | | |
| ~~4~~<br>~~11~~ | ~~MSB~~ | ~~new management identifier key~~ | | | | | | ~~LSB~~ |

The NEW MANAGEMENT IDENTIFIER KEY field ~~shall contain~~ specifies a new management identifier key.

If the override lockout timer managed by the access controls coordinator is not zero ~~non-zero~~, the access controls coordinator's states shall not be altered, ~~then~~ the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST~~,~~ and the additional sense code shall be set to INVALID FIELD IN CDB.

If the override lockout timer managed by the access controls coordinator is zero, then the access controls coordinator shall ~~reset~~ replace the current management identifier key with the value in the to the NEW MANAGEMENT IDENTIFIER KEY field ~~value in the parameter data. The device server shall respond with GOOD status~~.

### 7.2.9 REVOKE PROXY TOKEN service action ~~(Optional)~~

The ACCESS CONTROL OUT command with REVOKE PROXY TOKEN service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an initiator~~ to cancel all proxy access rights to a logical unit that were granted to third parties under the specified proxy token (see 5.99.5.2). ~~This is used in conjunction with the other PROXY-related service actions of the ACCESS CONTROL IN and ACCESS CONTROL OUT commands.~~ If this service action is not supported ~~by the access controls coordinator~~, the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, ~~and~~ the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

The format of the CDB for the ACCESS CONTROL OUT command with REVOKE PROXY TOKEN service action is shown in table t34 (see 7.2.1).

If access controls are disabled or if the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor eight ~~(8)~~, the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is eight ~~(8)~~, the parameter data shall ~~contain one eight (8) byte field specifying a Proxy Token and the device server shall always respond with GOOD status.~~ have the format shown in table t49.

**Table t49 — ACCESS CONTROL OUT with REVOKE PROXY TOKEN parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | | | PROXY TOKEN | | | | |
| 7 | | | | | | | | (LSB) |

If the PROXY TOKEN field does not contain a valid proxy token ~~in the parameter data is not valid, that is, not~~ associated with any logical unit at the access controls coordinator, ~~then~~ no further action is taken by the access controls coordinator. This shall not be considered an error.

If the proxy token ~~in the parameter data~~ is valid, ~~that is, associated with a logical unit at the access controls coordinator, then~~ the access controls coordinator shall take the following ~~additional~~ actions:

a) Invalidate the proxy token; and
b) Deny access to ~~that~~ the associated logical unit by any initiator whose rights were granted under that proxy token via ~~by~~ an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 7.2.11) according to the rules ~~of~~ stated in 5.99.6.

### 7.2.10 REVOKE ALL PROXY TOKENS service action ~~(Optional)~~

The ACCESS CONTROL OUT command with REVOKE ALL PROXY TOKENS service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an initiator~~ to cancel all proxy access rights to a specified logical unit that were granted to third parties under ~~all~~ any applicable proxy tokens (see 5.99.5.2). ~~This is used in conjunction with the other PROXY-related service actions of the ACCESS CONTROL IN and ACCESS CONTROL OUT commands.~~ If this service action is not supported ~~by the access controls coordinator~~, the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, ~~and~~ the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

The format of the CDB for the ACCESS CONTROL OUT command with REVOKE ALL PROXY TOKENS service action is shown in table t34 (see 7.2.1).

If access controls are disabled or if the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor eight ~~(8)~~, the ~~command shall be terminated device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is eight ~~(8)~~, the parameter data shall ~~contain one eight (8) byte field specifying a LUN value and the device server shall always respond with GOOD status.~~ have the format shown in table t50.

**Table t50 — ACCESS CONTROL OUT with REVOKE ALL PROXY TOKENS parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | | | LUN VALUE | | | | |
| 7 | | | | | | | | (LSB) |

If the LUN ~~value~~ in the LUN VALUE field is not associated to a logical unit to which the requesting initiator has non-proxy ~~any~~ access rights based on the contents of an ACE (see 5.99.2) or if the LUN value is based on a proxy token (see 5.99.5.2), ~~then~~ no further action is taken by the access controls coordinator. This shall not be considered an error.

~~If the LUN value is associated to a logical unit to which the requesting initiator has proxy access rights established on the basis of a Proxy Token, then no further action is taken by the access controls coordinator.~~

If the LUN value is associated to a logical unit to which the requesting initiator has non-proxy access rights, ~~that is, established on the basis of an entry in the ACL~~, ~~then~~ the access controls coordinator shall take the following additional actions:

a) Invalidate all proxy tokens associated to the logical unit specified by the LUN VALUE field ~~referenced by the LUN value~~;
b) Deny access to that logical unit by any initiator whose rights were granted under any of the invalidated proxy tokens ~~by a~~ via an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 7.2.11) according to the rules stated in 5.99.6.

### 7.2.11 ASSIGN PROXY LUN service action ~~(Optional)~~

The ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an initiator~~ to request ~~the access controls coordinator grant~~ access to a logical unit under the rights of a proxy token (see 5.99.5.2) and to assign that logical unit a particular LUN value for addressing by ~~that~~ the requesting initiator. ~~This is used in conjunction with the other PROXY-related service actions of the ACCESS CONTROL IN and ACCESS CONTROL OUT commands.~~ If this service action is not supported ~~by the access controls coordinator~~, the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN CDB.

The format of the CDB for the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action is shown in table t34 (see 7.2.1).

If the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor ~~sixteen (~~16~~)~~, the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is ~~sixteen (~~16~~)~~, the parameter data shall ~~contain the eight (8) byte Proxy Token associated with a logical unit followed by an eight (8) byte LUN value~~ have the format shown in table t51.

**Table t51 — ACCESS CONTROL OUT with ASSIGN PROXY LUN parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | PROXY TOKEN | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | LUN VALUE | | | | | | |
| 15 | | | | | | | | (LSB) |

If the contents of the PROXY TOKEN field are ~~is~~ not valid, then the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INVALID PROXY TOKEN.

NOTE 10 - If access controls are disabled, there ~~can be~~ ~~are~~ no valid proxy tokens~~. Consequently, in this state,~~ and the device server always responds with the ~~indicated~~ specified error information ~~status~~. ~~(~~This differs from the behavior of many other ACCESS CONTROL OUT service actions ~~is different from many other service actions~~ where the response is GOOD status ~~if~~ when access controls are disabled. The difference in behavior is intended ~~it is used~~ to inform the ~~initiator~~ application client that its request for the new LUN assignment failed.~~)~~

The LUN VALUE field specifies the LUN value the application client intends to use when accessing the logical unit described by the proxy token.

If the proxy token is valid but the access controls coordinator cannot assign the requested LUN value to the associated logical unit (e.g., ~~either~~ because the LUN value already is associated ~~to~~ with a logical unit ~~accessible to that~~ for the initiator, or because the LUN value ~~cannot be~~ is not a supported ~~as a valid~~ logical unit address), access rights shall not be granted, ~~then~~ the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to ACCESS DENIED - INVALID LU IDENTIFIER, and the SENSE-KEY SPECIFIC field shall be set as described for the ILLEGAL REQUEST sense key in 7.z.z. ~~Furthermore, the sense data shall be modified as follows. The SENSE-KEY SPECIFIC bit shall be set as described in 7.20.2 (of SPC-3, revision 0) with the FIELD POINTER field indicating the first byte of the requested LUN (as counted within the full parameter data) that differs from a value that may be supported by the access controls coordinator. Additionally,~~ The first ~~next~~ eight bytes of the additional sense bytes should contain ~~(if available) beyond the last byte of the FIELD POINTER may include~~ a suggested LUN value that the access controls coordinator ~~could~~ supports ~~for this proxy token. In this case, no new access rights are granted to the initiator.~~

If the proxy token is valid but the access controls coordinator has insufficient resources to manage proxy logical unit access ~~perform the requested action~~, ~~then~~ the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ~~of~~ ILLEGAL REQUEST and the additional sense code shall be set to ~~of~~ INSUFFICIENT ACCESS CONTROL RESOURCES.

If the proxy token is valid and the access controls coordinator has sufficient resources, ~~then the device server shall return GOOD status and allow~~ the initiator shall be allowed proxy access ~~for that initiator~~ to the referenced logical unit at ~~that~~ the specified LUN value ~~address~~.

### 7.2.12 RELEASE PROXY LUN service action (Optional)

The ACCESS CONTROL OUT command with RELEASE PROXY LUN service action ~~of the ACCESS CONTROL OUT command~~ is used ~~by an initiator~~ to release ~~remove a~~ proxy access ~~right~~ to a logical unit created with a proxy token (see 5.99.5.2) and the ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 7.2.11). ~~This is used in conjunction with the other PROXY-related service actions of the ACCESS CONTROL IN and ACCESS CONTROL OUT commands.~~ If this service action is not supported ~~by the access controls coordinator~~, the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST~~,~~ and the additional sense code shall be set to INVALID FIELD IN CDB.

~~This~~ The ACCESS CONTROL OUT command with RELEASE PROXY LUN service action should be used ~~by an initiator~~ when an initiator no longer requires the logical unit access rights granted under a proxy token ~~its access to that logical unit is no longer required under its proxy rights~~ (e.g., when a copy manager ~~server~~ has completed a specific third party copy operation ~~service~~ under ~~the~~ a proxy token). This allows the access controls coordinator to free any resources allocated to manage the proxy access ~~for that initiator~~.

The format of the CDB for the ACCESS CONTROL OUT command with RELEASE PROXY LUN service action is shown in table t34 (see 7.2.1).

If the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and ~~the device server~~ the command shall be completed ~~respond~~ with a GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor eight ~~(8)~~, the command shall be terminated ~~device server shall respond~~ with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is eight ~~(8)~~, the parameter data shall ~~contain the eight (8) byte LUN value as was used in the ASSIGN PROXY LUN service action~~ have the format shown in table t52.

**Table t52 — ACCESS CONTROL OUT with RELEASE PROXY LUN parameter data format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | (MSB) | LUN VALUE | | | | | | |
| 7 | | | | | | | | (LSB) |

The LUN VALUE field specifies a LUN value that was associated with a logical unit based on a proxy token using a ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action. If the LUN value was not assigned to a logical unit by an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action, the command shall be terminated with a ~~device server shall return~~ CHECK CONDITION status, ~~with~~ the sense key shall be set to ILLEGAL REQUEST and the additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

NOTE 11 - If access controls are disabled, there ~~can be~~ are no valid proxy tokens and therefore no LUN value could be assigned to a logical unit by an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action~~.~~ ~~Consequently, in this state,~~ so the device server always responds with the ~~indicated~~ specified error information ~~status~~. ~~(~~This differs from the behavior of many other ACCESS CONTROL OUT service actions ~~is different from many other service actions~~ where the response is GOOD status ~~if~~ when access controls are disabled. The difference in behavior is intended ~~it is used~~ to inform the ~~initiator~~ application client that the LUN value remains as a valid address for the logical unit.~~)~~

If the LUN value was assigned to a logical unit by an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action, the access controls coordinator shall disallow access to the logical unit at ~~this~~ the specified LUN value ~~address and the device server shall return GOOD status~~.

## F.8 – Protocol Specific Data

There is a proliferation of protocol specific command and parameter data that is being piled in SPC. Access controls is only the latest contributor to this onslaught.

The proposal here is to collect this data in a separate subclause in clause 8 by taking the following steps:

a) Create a subclause 8.x titled "Protocol specific parameters";
b) Create a subclause 8.x.a titled "EXTENDED COPY target descriptors" and move all the protocol specific target descriptors from their current location to 8.x.y;
c) Create a subclause 8.x.b titled "Mode pages" and move the Disconnect-Reconnect mode page plus the two protocol specific mode pages to 8.x.b; and
d) Create a subclause 8.x.c titled "Access controls TransportIDs" containing the subclauses shown here as 8.99.99…

### ~~7.1.1 Access Identifier types and lengths~~

~~Access identifiers are used in conjunction with access controls (see and specifically a.b.c) to identify an initiator or initiators for the purpose of granting, revoking or reporting on access rights. Access identifiers are specified in parameter data with an IDENTIFIER TYPE code and ACCESS IDENTIFIER field as defined in Table 33, as well as with a length field.~~

~~Table 36: IDENTIFIER TYPE and ACCESS IDENTIFIER values.~~

| ~~Code~~ | ~~Description~~ | ~~Length (bytes)~~ |
|---|---|---|
| ~~00h~~ | ~~AccessID~~ | ~~24~~ |
| ~~01h~~ | ~~TransportID~~ | ~~24~~ |
| ~~02h-7Fh~~ | ~~Reserved~~ | ~~n/a~~ |
| ~~80h-FFh~~ | ~~Vendor-specific~~ | ~~VS~~ |

~~The specification of the AccessID within the ACCESS IDENTIFIER field is given in 0.0.39. The specification of the TransportID within the ACCESS IDENTIFIER field for parallel SCSI initiators is given in 0.0.40 and for initiators using the SCSI over Fibre Channel protocol in 0.0.41. Other SCSI protocol standards may specify the structure of the TransportID and its description within the ACCESS IDENTIFIER field.~~

~~The TransportID format shall have a value in Byte 0 which uniquely identifies the transport protocol. Table 37 specifies the value for the parallel SCSI and SCSI over Fibre Channel protocols.~~

~~Table 37: Protocol identifiers for TransportIDs (Byte 0)~~

| ~~Protocol~~ | ~~TransportID Byte 0~~ |
|---|---|
| ~~parallel SCSI~~ | ~~00h~~ |
| ~~SCSI over Fibre Channel~~ | ~~01h~~ |
| ~~Other~~ | ~~Reserved~~ |

### ~~7.1.2 AccessIDs~~

~~The format of the AccessID within the ACCESS IDENTIFIER field in parameter data is described in Table 38. There are sixteen (16) bytes of significant data in this structure.~~

~~Table 38: AccessID data structure~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~<br>~~15~~ | ~~MSB~~ | | | ~~accessid~~ | | | | ~~LSB~~ |
| ~~16~~<br>~~23~~ | ~~Reserved~~ | | | | | | | |

# 8.99 Protocol specific parameters

## 8.99.1 Protocol specific parameters introduction

Some commands use protocol specific information in their CDBs or parameter lists. This subclause describes those protocol specific parameters. The descriptions in this subclause may be general, giving an overview to the protocol specific parameters as applied to all protocols or the descriptions may be specific to usage in a specific protocol. Each description includes a discussion of its scope.

In all cases, protocol specific parameter descriptions in a protocol standard (see 3.x.y) supersede descriptions for the same parameters in this standard.

## 8.99.99 Access controls TransportID access identifiers

### 8.99.99.1 TransportID introduction

TransportIDs (see table 53) are a type of access identifier (see 5.99.2.2) used in ACL ACEs to allow logical unit access to an initiator based on a protocol specific initiator device name, initiator port identifier, or initiator port name belonging to that initiator.

**Table t53 — TransportID format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PROTOCOL CODE | | | | | | | |
| 1 | Protocol Specific Data | | | | | | | |
| 23 | | | | | | | | |

The PROTOCOL CODE field (see table 54) identifies the protocol to which the TransportID applies.

**Table t54 — Protocol codes**

| Protocol Code | Protocol | Protocol Standard | Reference |
|---|---|---|---|
| 00h | SCSI over Fibre Channel | FCP-2 | 8.99.99.2 |
| 01h | Parallel SCSI | SPI-4 | 8.99.99.3 |
| 02h-FFh | Reserved | | |

### 8.99.99.2 TransportIDs for initiators using SCSI over Fibre Channel

A Fibre Channel TransportIDs (see table 55) is a type of access identifier (see 5.99.2.2) used in ACL ACEs to allow logical unit access to a FCP-2 initiator based on the world wide unique initiator port name belonging to that initiator.

**Table t55 — Fibre Channel TransportID format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PROTOCOL CODE (01h) | | | | | | | |
| 1 | (MSB) | | | Reserved | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | | | WORLD WIDE NAME | | | | |
| 15 | | | | | | | | (LSB) |
| 16 | (MSB) | | | Reserved | | | | |
| 23 | | | | | | | | (LSB) |

The WORLD WIDE NAME field shall contain the port World Wide Name defined by the Physical Log In (PLOGI) extended link service, defined in FC-FS.

A Fibre Channel TransportID allows the initiator specified by the world wide name access to the logical units described in an ACE (see 5.99.2).

~~The format of the TransportID within the ACCESS IDENTIFIER field in parameter data for the FCP protocol is described in Table 39.~~

~~Table 39: TransportID for FCP.~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~01h~~ | | | | | | | |
| ~~1~~<br>~~7~~ | ~~Reserved~~ | | | | | | | |
| ~~8~~<br>~~15~~ | ~~MSB~~ | | | ~~wwportname~~ | | | | ~~LSB~~ |
| ~~16~~<br>~~23~~ | ~~Reserved~~ | | | | | | | |

~~The WWPORTNAME field is the Worldwide_Name of the Fibre Channel port.~~

### 8.99.99.3 TransportIDs for initiators using a parallel SCSI bus

A parallel SCSI bus TransportIDs (see table 56) is a type of access identifier (see 5.99.2.2) used in ACL ACEs to allow logical unit access to a SPI-4 initiator based on the SCSI address of an initiator and the SCSI target device relative port through which the initiator accesses the SCSI target device.

**Table t56 — Parallel SCSI bus TransportID format**

| Bit<br>Byte | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| 0 | PROTOCOL CODE (00h) | | | | | | | |
| 1 | Reserved | | | | | | | |
| 2 | (MSB) | SCSI ADDRESS | | | | | | |
| 3 | | | | | | | | (LSB) |
| 4 | (MSB) | RELATIVE PORT IDENTIFIER | | | | | | |
| 7 | | | | | | | | (LSB) |
| 8 | (MSB) | Reserved | | | | | | |
| 23 | | | | | | | | (LSB) |

~~The format of the TransportID within the ACCESS IDENTIFIER field in parameter data for the parallel interface is described in Table 40.~~

~~Table 40: TransportID for SPI.~~

| ~~Byte~~ | ~~Bit~~ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ~~7~~ | ~~6~~ | ~~5~~ | ~~4~~ | ~~3~~ | ~~2~~ | ~~1~~ | ~~0~~ |
| ~~0~~ | ~~00h~~ | | | | | | | |
| ~~1~~ | ~~Reserved~~ | | | | | | | |
| ~~2~~<br>~~3~~ | ~~MSB~~ | ~~scsi address~~ | | | | | | ~~LSB~~ |
| ~~4~~<br>~~7~~ | ~~MSB~~ | ~~relative port identifier~~ | | | | | | ~~LSB~~ |
| ~~8~~<br>~~23~~ | ~~Reserved~~ | | | | | | | |

The SCSI ADDRESS field ~~indicates~~ specifies the SCSI address (see SPI-4) of the initiator.

~~**AUTHOR'S NOTE:** *The SCSI Address is defined in the glossary of SPI-4 (rev 05) in item SPI-3.1.87.*~~

The RELATIVE PORT IDENTIFIER ~~shall indicate~~ field specifies the four-byte binary number identifying a specific port in the SCSI target device relative to other ports. The relative port identifier value shall be one of the values returned in the Device Identifier VPD page (see 8.z.z). ~~(see Table 175 of SPC-3 rev 0). The relative port identifies a SCSI domain in which the SCSI ADDRESS is a unique identifier of a SCSI device.~~ If the RELATIVE PORT IDENTIFIER does not reference a port in the device, the TransportID is invalid.

In order for a parallel SCSI bus TransportID to allow access to the logical units described in an ACE (see 5.99.2), an initiator having the specified SCSI address shall access the SCSI target device via the port specified by the relative port identifier.