Date: April 26, 2000

To: T10 Committee (SCSI)

From: Jim Hafner (IBM) (hafner@almaden.ibm.com)

Subject: A Detailed Proposal For Access Controls

ABSTRACT:

A SAN (storage area network) is a network environment where multiple hosts machines (clients/initiators) have access to a collection of storage devices (targets). Unless there is significant collaboration between the initiators, it is desirable in this environment, to partition, fence or otherwise restrict access to some storage devices by different hosts. The current SAN protocols (either at the transport layer or in the SCSI layer) are not well-suited to this purpose.

In this proposal, we detail new SCSI commands and target actions to implement access control management. Two new commands are proposed that allow configuration (Data-Out) and reporting (Data-In) of access control management functions at the target. The new commands and actions are not restricted to storage devices but are applicable to any target.

This and earlier revisions reflect comments, questions and suggestions from folks at LSI Logic, Sun Microsystems, Adaptec, Compaq and others at IBM.

Revisions 5-8 were based on extensive discussions between Ralph Weber (ENDL Technologies), David Chambliss (IBM) and the author. The general framework of the model is joint work. But the author is responsible for the content of this document, particularly for inconsistencies, incompleteness, errors, or blunders.

In particular, revision 5 attempted to merge the requirements presented in 00-123r0 (for what has been dubbed LUN Mapping) and the additional features of 99-245r4.

Revision 8 includes mostly an extensive rewrite of the model clause (and changes to the command clauses that derive from changes in the model clause). A number of minor changes to the model are included as well.

This revision has no specific provision for access controls on subcomponents.

1.0 Introduction

A SAN (storage area network) is a network environment where multiple hosts machines (clients/initiators) have access to a collection of storage devices (targets). Unless there is significant real-time collaboration between all the initiators, it is desirable in this environment, to partition, fence or otherwise restrict access to some storage devices by different hosts. The current SAN protocols are not well-suited to this purpose of access control management.

In our view, access controls should have the following properties:

a) they should be enforced at the target;

b) they should be granted to a host (i.e., at the OS-image or virtual machine level) and not to particular initiators (or ports or HBAs) within a host if at all possible;

c) they should be configured by some application client that is responsible for overseeing access controls over the entire SAN;

d) a configuration of access controls should not be associated with the particular initiator from which the configuration command was sent.

The last three points imply that SCSI reservations are inadequate to the task unless there is a single (real-time) application client coordinating reservations for *all* initiators in the SAN simultaneously. Such an application in a complex, multi-OS, multi-initiator environment would be expensive and difficult to manage.

To enable the protection required for access to devices in a simpler and easier to manage way, we propose a new SCSI-based protocol for access controls. This protocol is independent of the transport layer and is suited for any SAN environment whose higher level protocol is SCSI (e.g., FCP).

A general scenario is the following. A client application (what we call the Partition Access Manager or PAM[1]) has knowledge of all the initiators and target devices on the SAN. PAM may instruct a given target device to restrict access to some or all of its logical units by all initiators except those from some small set. Such a set might be a single host. Within the set, data integrity, locking, etc., is coordinated by existing protocols (like reservations) via a separate application client operating within the scope of this group. One might say that such a set is a “shared access group”. Hosts outside this group are denied (most) access to the device. In particular, these hosts may not preempt a reservation, issue read/write commands and the like. (Provisions for quality of service or resource allocations within a “shared access group” are outside the scope of this proposal.)

Note the following features of this scenario. PAM need only have one in-band communication channel to the target devices. PAM does not need to have any active presence on all the initiators, because the configuration commands are initiator independent. Furthermore, access restrictions are enforced at the target devices. This means that new hosts added to the SAN have no access to restricted targets unless expressly added by PAM. Also, hosts need have no special application client running in order to “fence” them from target devices to which they should not access or to gain access to devices to which they have been granted access.

The proposal is applicable to any kind of target, not just storage devices. Resource requirements at the target vary so that even limited function devices such as disk drives themselves might be capable of implementing these functions. However, it is more likely that larger devices such as controllers, devices with an embedded controller, medium changers, intelligent bridges (e.g., FC to SCSI) and the like would implement these functions.

There are two new commands with different service actions proposed. There is a Data-In command typically used to query various status information of the target with respect to access control functions and a

1.PAM is not part of the proposed standard, nor is it necessarily a real application. Mainly it is a pseudonym for the management application overseeing access controls for the SAN. It may be instantiated by a real application or instantiated more generally by the use of the defined protocol by users.

Data-Out command typically used to configure different kinds of access controls. These are detailed in later sections. However, use of configuration commands are limited with respect to application clients or initiators. Initiators with access to a device have the right to issue proxy rights to other third party initiators without PAM's direct intervention. On the other hand, PAM's configuration tools (MANAGE ACL service action) may only be used by an application client (namely PAM) that shares a key with the target. This key identifies PAM as the originator of the command independent of which initiator she uses for command delivery. The key is maintained as part of the access control information of the target and must be preserved through power cycles.

A model for override of the target's key (in the unlikely event that PAM forgets it) is provided. This model allows for a lockout state where the key overrides are not allowed and an unlocked state where a simple override may occur with a defined command and service action. The lockout state may be easily maintained by PAM through management of a countdown timer; restart of the timer requires little overhead and does not even require PAM's "key". The initial value of the timer is configured by PAM.

Hosts (or OS-images) may be identified by a new AccessID as defined in this proposal. The reasons for the new identifier are the following. First, the new AccessID is transport independent and so is applicable to all current and future transport protocols. Second, (as noted above) access rights are naturally associated with the host machine (or virtual machine), not the individual initiators (ports/HBAs) on that machine. Transport layer identifiers, either transitory (e.g., FC N_Port) or persistent world wide identifiers (e.g., FC World Wide Nodename) are cumbersome and inadequate. Because they are bound to the given HBA within a machine, they are portable. This would require PAM to maintain continual knowledge of host hardware configuration simply to manage access rights. However, for additional function, the design contains provisions for transport-layer as well as vendor-specific identifiers.

The intent of the AccessID is to assign a permanent identifier to a given host machine (actually OS-image) without regard to the number of ports/HBAs on that host or any actions that change the hardware configuration of the machine. This makes management by PAM of the target's access controls much simpler. But it also implies requirements on the part of target to maintain associations between the AccessID and a given host's initiator port or ports. These requirements are similar to but in some cases less restrictive than those already required by reservations.

Though AccessIDs create a new identifier name space that PAM must manage, it is our opinion that the gains in simplicity, stability and transport independence outweigh this concern.

What follows is a detailed description of the new commands and target requirements. Section 2.0 provides an informative description of the model, raises some design questions and issues and documents the revision history. The remaining part of the document is proposed normative changes to existing standards documents. Section 3.0 contains proposed changes to the glossary and acronyms clauses of SPC-3. Sections 4.0 is proposed as an additional sub-clause in the model clause of SPC-3. Sections 5.0 and 6.0 are proposed as additional command definition clauses to SPC-3 and Section 7.0 proposes an additional subclause to SPC-3, clause 8. The appendices propose additional small changes to SPC-3 and other standards documents.

AUTHOR'S NOTE: *AUTHOR'S NOTEs are intended to generate small questions and expose small issues for possible further action. Ideally, later revisions of this document will have these issues addressed and the notes removed. In any case, they should not be included in the final editorial changes included in SPC-3. Larger issues are listed in the next section.*

2.0 Additional major issues or questions[1]

2.1 The new model and open issues

There are significant changes to the basic access controls model introduced in revision 5 and continued here. We summarize the new model in more detail in what follows, but begin by highlighting a few of the major differences with the old model.

The earlier drafts (revision 4 and earlier) all shared the “visible but inaccessible” approach to access denial. That is: all logical units are visible to all initiators (meaning that they are discoverable under INQUIRY and REPORT LUNS) but non-privileged initiators are denied service (in particular I/O service) by specific CHECK CONDITIONS. This new draft replaces the “visible but inaccessible” model with what has been dubbed LUN Mapping. This has two features. First, it “hides” logical units that are not accessible to a given initiator. So, to such an initiator INQUIRY to some logical units will report “no device present” and REPORT LUNS will only show a set of LUN values representing a subset of the complete set of logical units on the target device. The second feature of LUN Mapping is that the LUN values reported in REPORT LUNs are initiator-specific. That is, for each initiator a given LUN will only be a pointer to an specific logical unit and that pointer is a function of the initiator. Thus, the same (shared) logical unit may be addressed by one initiator at LUN1 and by another initiator at LUN2. A consequence is that LUN values are no longer global addresses for specific logical units within a target. This complicates a number of shared functions (such as third party copy operations). This “mapping” function however is seen as a functional requirement.

Because of the changes required for LUN Mapping, the proxy model has changed significantly. This is discussed in more detail in 2.1.2.

AUTHOR’S NOTE: *This section refers to LUN Mapping. However this term no longer (as of revision 8) appears in the model clause. Language that used to refer to an initiator’s LUN Map has been changed to more accurately reflect on the ACL or on an initiator’s rights derived from the ACL. However, we still use the term here, as it best encapsulates the concepts. A fresh and critical reading of the model clause is probably required to make sure that the concepts of LUN Mapping are adequately described there.*

The new model begins with the following assumptions and requirements:

a) PAM knows best what the LUN Map should look like for a given initiator; she needs this ability to minimize the “move LUN” problem when access controls rights are changed for a given initiator (this is changed from revision 5);

b) most hosts have the ability to handle gaps in there LUN list (this is changed from revision 5); PAM assumes the responsibility for this if required;

c) host resources for never-accessible logical units should not be wasted;

d) some hosts require access to a specific logical unit at LUN 0 (for boot?) (this is unchanged from revision 5 but the implementation of this is different);

e) facilitate third-party operations in a simple way (proxy);

f) need an interlock with hosts to assist them with enrollment requirements;

g) LUN Map changes should be minimized and managed so that data integrity is protected and host efforts to recover from LUN Map changes are minimized (this is better facilitated by giving PAM ownership of the LUN Map);

h) provide a simple mechanism to both allow for the override of PAM’s shared key, under controlled conditions, and to easily manage these conditions.

With these in mind, the model has the following characteristics.

1.This section is primarily to discuss the model, raise discussion points and to log changes to the various revisions; it is not part of the proposed standard.

First, PAM tells the target device to grant access to a particular logical unit to a given initiator under a specific LUN value (in other words, PAM instructs the target to create a LUN Map entry of a particular type for the specified initiator).

An initiator may be identified to the target by either a TransportID (available at connection time, say, FC-login) or by AccessID (available only after an enrollment action by the initiator). Since the latter may only come after the connection, the target will always map TransportID-accessible logical units before mapping AccessID-accessible logical units. Furthermore, an initiator (such as a copy manager) may get access (have its LUN Map changed) through proxy functions.

Because of the independence of the LUN Maps defined for TransportID and AccessID, it is possible that PAM may instruct the target to create a LUN Map that can not be instantiated. There are two possibilities that need to be dealt with:

a) PAM instructs the target to map a specific LUN value to one logical unit under a TransportID and to another under an enrolled AccessID;
b) PAM instructs the target to map a specific logical unit by one LUN value under TransportID and by another LUN value under an enrolled AccessID.

Note that neither of these conflicts may be detected at the time PAM instantiates the configuration (that is, when PAM issues the MANAGE ACL service action), without additional functionality; these conflicts might only be detected at the time of an enrollment. Consequently, the target needs a rule to handle conflict resolution on the fly and also needs a means to report that to PAM on request. To handle this, we require that the target maintain some log of conflict resolutions (the log is limited by the resources of the target in implementation dependent ways). The minimal log is a counter of the number of conflict resolutions that the target had to handle. The log may also contain a list of conflicts that occurred and were resolved by the target (until log resources were exhausted). The log may be queried by PAM and cleared by PAM via specific service actions. Additional log features are also specified.

The TransportID and AccessID portions of the LUN Map have the following properties:

a) LUNs in these portions normally appear during system (host) startup and remain unchanged from one boot to the next;
b) more than one LUN may appear in either of these portions as a result of a single action on the part of the initiator (e.g., PLOGI or enrollment);
c) automated loss of access (e.g., LIP) may be corrected by repeating the PLOGI or enrollment;
d) only one LUN value shall be assigned to a logical unit under both TransportIDs and AccessIDs (that is, for logical units accessible under both TransportIDs and AccessIDs, the mapping is one-to-one).

The Proxy portion differs as follows:

a) LUNs in this portion do not appear normally during system startup and may change dynamically during the life of the system;
b) exactly one new LUN appears in this portion as the result of a single action by the initiator (ASSIGN PROXY LUN service action);
c) automated loss of access will not be corrected by the same mechanisms that work for TransportIDs and AccessIDs.
d) multiple LUN values assigned under proxy may reference the same logical unit.

Note that a logical unit may be addressable by a given initiator at a single LUN value in the TransportID or AccessID portion and/or via one or more LUN values in the proxy portion.

The TransportID portion of a LUN Map for an initiator after a target reset will be restored. The AccessID portion may not (in an implementation-specific manner) be restored. But such an initiator will easily detect the failure, recognize the potential reasons and take the necessary action (enroll) to restore access.

We assume three possible states for an initiator with respect to a target:

a) “enrolled” - in this state, LUN Map contains entries for logical units to which the enrolled AccessID has been granted access;

b) “pending-enrolled” - in this state, that is the transition state from enrolled because of automated loss of access (e.g., LIP) or PAM-initiated event (e.g., Flush in MANAGE ACL service action), the AccessID accessible logical units stay in the LUN Map, but are “inaccessible” with status and sense indicating that the enrollment was invalidated;

c) “not-enrolled” - in this state, the target has no association of an AccessID for the initiator; the initiator’s LUN map contains only references to logical units for that initiator’s TransportID.

Simply, the first enrollment action by an initiator merges the LUN Map entries for the enrolled AccessID into the existing LUN Map and puts the initiator in the “enrolled” state. Events (outside of the initiator’s control) may cause the initiator to go into the pending-enrolled state. The LUN Map doesn’t change, but commands to the affected logical units are failed with sense data sufficient to trigger the initiator to re-enroll. This would put the initiator back into the enrolled state. The initiator switches between these two states under normal operation.

An initiator may go into the not-enrolled state under two events. The first is by its own actions (CANCEL ENROLLMENT). This should be used by an initiator prior to shutdown so the target may free up enrollment resources. This is not required, however. The second event is an action by PAM that causes a change in the LUN Map for that initiator. Such events are (or should be) rare but they have the potential to adversely affect the host and its data. This scenario is described in 2.1.1.

We propose the existence of a new entity in a device. We call this the “access controls coordinator”[1]. This entity is the repository of the access controls data, the coordinator of access rights, the handler of enrollments and the builder of LUN maps. This entity spans multiple logical units as well as spans all the ports in a multi-ported device. In effect, it creates an interface between the ports and the logical units for the purpose of restricting access, and managing the LUN Maps for initiators. This entity is addressable only (or “should” be only) via LUN 0. [We’ve allowed for access control commands to go to any LUN, to facilitate the difficulties of host/OSs that do LUN offsetting or LUN mapping internally.]

2.1.1 Host/target/PAM interlock for LUN Map changes

Actions by PAM to the ACLs for a given TransportID or AccessID may cause a change in the LUN Map for some initiators. If not coordinated carefully with the affected initiators, this may have undesirable effects on the user data. E.g., if a host has a string of I/Os enqueued (at the host) addressed to a LUN value and intended for a particular logical unit, and PAM changes the LUN Map so that this LUN value no longer addresses the same logical unit, the enqueued I/Os will (without a specific interlock) go to the wrong logical unit, thereby both corrupting the data on the newly addressed logical unit and the hosts view of that data consistency.

Note that this issue arises only if a LUN value “moves” to a new logical unit; not if the LUN value is “added” to the LUN Map or if the LUN value is “deleted” from the LUN Map. For the added case, there would be no active I/Os, even though the initiator will need to take some action to “discover” the new logical unit. For the “deleted” case, all I/Os will fail with “logical unit not supported”, and no data is transferred.

To address this problem, we first postulate that PAM will never take this sort of action unless she has sufficient knowledge that such risks to data integrity have been minimized. This might mean making sure that the affected hosts are shutdown (or quiesced) and that sufficient rediscovery of the LUN Map by the host will correct the problem.

1.Alternatives to “coordinator” are “enforcer” or “manager” or “enforcement manager”. Enforcer is an unpleasant word and manager might be confused with PAM, so we chose coordinator.

Second, a tight interlock between PAM and the host is actually only required for that portion of the LUN Map in the TransportID portion. The AccessID portion may be handled to a certain extent by relying on the enrollment process itself (see below). TransportIDs will (should) only be used in a couple of cases. In the first case, if the host does not participate in the access controls protocol (e.g., a legacy system). In this case, PAM will need a direct interlock with the host. In the second case applies to perhaps a very limited set of logical units (e.g., for boot devices) for hosts that do participate in the enrollment protocol; most of their LUN Map should be in the AccessID portion. In this case, there is less likelihood of an action by PAM affecting the LUN Map, so less need for direct PAM/host interlock.

Third, and most important, since PAM owns the LUN Map, she may implement a policy that significantly reduces or eliminates this problem.

For the AccessID portion, we propose the following host/target interlock. If an ACL entry for an AccessID is replaced by PAM, any initiator enrolled under that AccessID has its enrollment cancelled (moved to the not-enrolled state); all formerly addressable logical units become “not supported”. This error message should be sufficient notification to the host that something dramatic has happened (either the map changed or the target reset). In either case, the host should suspend IO, re-enroll and redrive the LUN-discovery process (at the affected target) to clean up its internal references to logical units.

Additionally, we give PAM the ability (with a bit in parameter data) to request that the target not to cancel the enrollment. The burden is on PAM to check that the run-time behavior of a host whose enrollment is not cancelled is safe. However, the target may cancel the enrollment if it chooses to in any case; this gives the target vendors the ability to supersede PAM’s judgement, if they chose.

2.1.2 The proxy model

In a previous revision, an initiator would grant a third party (based on an identifier for that third party) proxy access to a logical unit to which the granting initiator already had access. This had a couple of limitations (e.g., a copy manager given proxy access couldn’t farm out part of its job to another party). In the presence of LUN Mapping, however, the problem is more pronounced in that there is no longer global addressing of logical units (by LUN). Consequently, if initiator A wanted initiator B to have access, A would need to grant proxy right (at the target) to B, somehow find out what LUN got generated for that logical unit in B’s LUN Map and use this value when requesting services from B (e.g., in EXTENDED COPY target descriptors). In this revision, we change the proxy model significantly. An initiator, instead of granting access to a third party by initiator identifier, requests a proxy token from the target for a specific logical unit, passes the token to the third party, and that third party uses the token to request a LUN value for that logical unit.

This has the following advantages:

a) global addressing of logical units by LUN is no longer required in the presence of a LUN Map;

b) proxy tokens may be forwarded from one third party to another;

c) multiple proxy tokens may be used for the same logical unit;

d) each token held by a third party may be used to assign a separate LUN value (for the same logical unit); this allows the third party like a copy manager to separate the required tasks by LUN);

e) separation of proxies between initiators sharing a given logical unit; e.g., if initiator A and B have forwarded proxy tokens for a shared logical unit, they may invalidate their own tokens without affecting the other initiator.

We have proposed changes to the EXTENDED COPY target descriptors to include Proxy Tokens as handles or references to logical units.

We propose that proxy tokens may be invalidated by an initiator that knows the proxy token, by any initiator with PAM-granted access to the logical unit (to invalidate all proxy tokens for a given logical unit) or by PAM either by individual proxy token or for all proxy tokens at a target.

In revision 5, we mapped out two variants of a protocol for the third party to get a LUN value for a Proxy Token. The preferred method is that the third party initiator specifies a preferred LUN value when requesting the access to the logical unit (that is, the initiator asks “may I have LUN=x for Proxy Token=t?”). The alternative is the third party initiator requests the target generate a LUN value for the token (that is, “what LUN value may I have for Proxy Token=t?”). In revision 6, we stabilize on the preferred method. Unfortunately, this model may open up the “twenty questions” problem. To alleviate this, we’ve specified that the target include an alternative LUN value embedded in the sense data of the failed request. See 6.2.11.

2.1.3 Override of Management Identifier Key

This design requires that the Management Identifier Key be used by an application client (PAM) whenever it wants to change access controls at a target. We expect that this application will reliably maintain this information and do not expect that PAM will “forget” the key. However, we need to allow for some mechanism to override the existing key, just in case. Many alternatives have been proposed. In this revision (rev 7) we propose the following model, that incorporates the notion of a “state machine” (to allow override only if the device is in a special state) and “fingerprints” (to record attempts, both failed and successful, to override the key). The curious thing about this model is that the default state is to allow unrestricted override; only in a changed state is the override prevented. Maintaining a device in this special state is simple and requires very little overhead.

The access controls coordinator maintains a configurable timer (a non-negative integer) that decrements approximately once per second (until it reaches zero).

If the timer is zero, the override attempt (via the OVERRIDE MGMT ID KEY service action) shall succeed. If the timer is not zero, any attempt to override the key shall fail. Furthermore, all attempts to override the key are logged with the TransportID of the initiator sending the service action, the state of the timer and additional implementation-dependent data (e.g., real-time clock, if available).

The timer is managed with the MANAGE OVERRIDE LOCKOUT TIMER service action. This has two possible functions:

a) reset the initial (starting) value for the timer, check the current value of the timer, and restart the timer (this action requires validation with the Management Identifier Key);

b) restart the timer (this does not require validation with the Management Identifier Key).

In other words, PAM (and only she) has the ability to define the initial state of the timer and get the current value for the timer. Any initiator may restart the timer.

An initial setting of zero for the timer effectively disables the timer.

This enables PAM to easily maintain all the devices she manages in the override locked state (make sure all the timers are positive) in a manner consistent with objectives and policies of a particular deployment. There is very little overhead for her to maintain this state for all the target devices.

If for any reason PAM forgets the keys she uses to manage a device (or all devices), she may allow the timer to lapse, and then issue the override.

We are requiring that the timer be restarted (reset to its initial value and resume decrementing once per second) under each of the following conditions:

a) on demand, with successful completion of MANAGE OVERRIDE LOCKOUT TIMER;

b) target resets and power-cycles.

Furthermore, we require that the initial setting of the timer be persistent and non-volatile.

We do not require a real-time clock or even a very accurate timer, though we do require a minimum level of clock accuracy. PAM could (approximately) measure the timer’s internal accuracy (deviation from once per

second decrements) by experiment, namely, querying the current value of the timer (with REPORT OVERRIDE LOCKOUT TIMER service action).

2.1.4 Access controls log

Revision 6 specified a somewhat incomplete access controls log. That version contained only a log for recording LUN Mapping conflicts that arise because an initiator's AccessID LUN Map was incompatible in some respect with the LUN Map created for that initiator's TransportID.

Revision 7 expands on the log and defines its behavior more carefully.

Revision 8 changes the terms for LUN Mapping conflict to ACL LUN conflict.

There are three portions to the log, for recording different events. These are: (a) attempts to override the Management Identifier Key, (b) use of commands with invalid key and (c) ACL LUN conflicts (as before). Each portion of the log minimally contains a counter of the number of such events and optionally additional information about each occurrence of these special events.

We have defined two service actions to deal with the log, one in the ACCESS CONTROL IN command to report on the log and one in the ACCESS CONTROL OUT command to clear the log. Note that the "override keys" portion of the log cannot be cleared. Reading this portion of the log does not require knowledge of the Management Identifier Key. If it did, someone could override the key and either clear the log (erasing the "fingerprints") or prevent the real PAM from reading the log (hiding the fingerprints).

We intentionally did not put the log under the LOG SENSE/SELECT command for a number of reasons:

a) these commands are blocked by access controls so are not available to PAM unless she (or an agent) have direct access rights to LUN 0 (i.e., the logical unit through which the access controls coordinator is addressable);

b) these commands are available to anyone with access rights (not just PAM, assuming she had granted herself access) and some of the data (e.g., AccessID's) is information that ideally should not be readily accessible to all initiators;

c) not using these commands better encapsulates the changes to SPC-3 for access controls.

The last point here is not really that strong a point as there are many places where this proposal impacts other parts of the standard. However, in most of those cases, the changes are minimal (one paragraph and a field change in some parameter data); changes to LOG SENSE/SELECT for this purpose would be much more intrusive.

On the other hand, the other two points are stronger arguments in favor of encapsulating this part of the model within key-validated (with exceptions) access control commands.

2.1.5 OPEN QUESTIONS

As far as the author is concerned, all issues are closed. However, the reader may not share this opinion. The author is open to comments and suggestions for changes both to the basic model, the resource requirements, specifics in parameter data structures and editorial issues.

2.2 Access controls on sublogical unit entities

Revision 4 and 5 have no notion of access control granularity below the logical unit level, though earlier revisions did. There may be a reason in the future to extend access controls to sublogical unit entities. In one context (medium changer) this might be elements. In the up-and-coming Object-based Storage Device model, this might be for access controls on Object Groups (this would provide some simple access controls without the need for complex encryption and authentication protocols - admittedly, this is not a complete solution to the long term objectives of OSDs, but might provide an interim solution).

It would be possible to extend the current model to allow such finer grained controls. The fundamental question however is what direction that may take. There are two alternatives:

a) Access grant to a logical unit (either by PAM or by proxy token) grants access only to the higher level entity and not any addressable subentity, unless specified otherwise.

b) Access grant to a logical unit (either by PAM or by proxy token) grants complete access to all sublogical unit entities, unless specified otherwise.

In the first case, we would need service actions and/or parameter structures to extend rights to sublogical unit entities. For example, give initiator A and B access to the logical unit, but they can't use "elements" X or Y within the logical unit. Then we expressly extend A's rights to X and B's rights to Y.

In the second case, we need service actions and/or parameter structures to limit rights. For example, give A and B access to the logical unit and they both can use X and Y. Then we expressly restrict A's access to only X and B's to only Y.

Revision 3 had a different, somewhat intermediate, model. In that version, explicit grant to a full logical unit gave rights to all subentities (similar to case two above); explicit grant to a subentity was limited to that subentity and the top level entity (similar to case one above). This doesn't quite work in this case, mostly because such semantics don't fit (in the author's opinion) very well with LUN Mapping syntax. The proposed options above separate the LUN Mapping service action/parameter data syntax from the subentity extend or restrict syntax.

For example, suppose we adopt case two. Initiator A has access to some or all of a logical unit and wants B to do some copy services for it, but limited to only a subset of A's access. A requests a proxy token from the target that is initially valid for the entire context of A's rights. A follows that request with a specific request to the target to limit the validity of the proxy token to a specific subset of its accessible subentities. A then forwards the proxy token to B with the assurance that B can't get access to that part of the logical unit not accessible to A and not included in the subset scoped by the proxy. (Similar syntax may be used by PAM, to first map a LUN value for a given logical unit and then restrict the range of validity of that access.)

The author has no particular preference for either model at this time.

2.3 Access Control Commands and Service Actions

Table 1 gives a summary list of the access control commands and their service actions.

TABLE 1. Access Control Commands and Service Actions

Code	Name	Type	Clause
ACCESS CONTROL IN (OPCODE 86h)			
00h	REPORT ACL	M	5.2.2
01h	REPORT LU DESCRIPTORS	M	5.2.3
02h	REPORT ACCESS CONTROLS LOG	M	5.2.4
03h	REPORT OVERRIDE LOCKOUT TIMER	M	5.2.5
04h	REQUEST PROXY TOKEN	O	5.2.6
05h-17h	Reserved		
18h-1Fh	Vendor-specific	V	
ACCESS CONTROL OUT (OPCODE 87h)			
00h	MANAGE ACL	M	6.2.2
01h	DISABLE ACCESS CONTROLS	M	6.2.3
02h	ACCESS ID ENROLL	M	6.2.4
03h	CANCEL ENROLLMENT	M	6.2.5
04h	CLEAR ACCESS CONTROLS LOG	M	6.2.6
05h	MANAGE OVERRIDE LOCKOUT TIMER	M	6.2.7
06h	OVERRIDE MGMT ID KEY	M	6.2.8
07h	REVOKE PROXY TOKEN	O	6.2.9
08h	REVOKE ALL PROXY TOKENS	O	6.2.10
09h	ASSIGN PROXY LUN	O	6.2.11
0Ah	RELEASE PROXY LUN	O	6.2.12
0Bh-17h	Reserved		
18h-1Fh	Vendor-specific	V	

The data in this table is included in two parts, one part in the ACCESS CONTROL IN command clause 5.2.1 and one part in the ACCESS CONTROL OUT command clause 6.2.1.

In revision 8, we extended the reserved range (at the cost of reducing the vendor-specific range).

2.4 Access Control Additional Sense Codes

Table 2 contains a list of the Additional Sense Code and Additional Sense Code Qualifiers relevant to access controls. The contents of this table, suitably modified for inclusion in SPC-3, may be found in Appendix D (Table 40).

TABLE 2. Access Control Additional Sense Codes and Qualifiers

ASC	ASCQ	Description	Function
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	04h	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	05h	ACCESS DENIED - INVALID LU IDENTIFIER	A LUN or default LUN value in a CDB field or parameter data is not valid.
20h	06h	ACCESS DENIED - INVALID PROXY TOKEN	The Proxy Token is not valid; it does not correspond to a logical unit.
20h	07h	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.

2.5 Changes from previous revisions

2.5.1 Changes for revision 3

A TransportID is defined for SPI devices.

The language concerning the effects of changes on access controls to commands already in the task manager has been clarified and simplified. It is modeled on the language from PERSISTENT RESERVATIONS.

There is a new OUT service action, RESET AC, that provides a Management Identifier Key validated reset function and a template for vendor-specific reset functions (that might provide an override mechanism for the Key).

There are additions to Table 8 of SPC-2 defining the device server's actions in the presence of reservations when access control commands are issued.

The table of new ASC/ASCQs for access controls has been updated with specific values, consistent with the proposal 99-314r1.

The model for access controls on elements (or more generally on subcomponents of the logical unit) has be significantly redone. There is a definition for an "access controllable component (ACC)" and revised specification of an initiator's access rights when granted access only to such a component. Also, there is now only one ACL at the device server (not one per ACC and the logical unit) and so there is only one ACL Enabled or Disabled state for the device. This was done both to simplify the model (and hopefully clarify it) and to enable a simple evolution to the next revision where this model is deleted (so access controls are only defined at the full logical unit). At the moment there is no driving force behind having this finer granularity, which is why it will be excised from the next revision. We are archiving this revision with the revised model in the event that a future need arises for access controls at a granularity below the full logical unit.

For example, the model described here could work with elements of medium changers as originally expected or it could be applicable to access controls on object-groups as might be defined in the Object-based Storage Device proposal currently under review (99-315r0).

This change in the “element” model removes a certain functionality, namely, of disabling access controls on specific ACCs within the device while still maintaining access controls at the full logical unit. There are two possible approaches to this. One is to have a configuration command that can change the classification of a subcomponent from an ACC to a non-ACC component (though this might be hard to define carefully). A second approach is to define a “universal AccessID” that all initiators are automatically enrolled under (so a sort of wild-card AccessID). Granting access to this universal AccessID would be functionally equivalent to disabling access controls at the specific ACC.

Other wording changes of an editorial nature are included here as well.

2.5.2 Changes for revision 4

The model for access controls on subcomponents of the logical unit has been removed.

2.5.3 Changes for revision 5

This is a major revision, and so has substantial changes.

There have been some name changes. E.g., Manage ACL Key is now called the Management Identifier Key. The RESET AC service action has been renamed to DISABLE ACCESS CONTROLS. Many service actions have been changed, added or removed. The parameter data for some commands has changed as well. The set of required ASC/ASCQs has also changed. These changes are detailed in the sections that follow.

There is an additional clause (Appendix D.2) that proposes changes to the EXTENDED COPY command’s target descriptors to accommodate proxy tokens.

The SCSI Address field in TransportID for SPI-4 has a reference to the glossary for this name.

The TransportID for FCP-2 has been changed. N_PortID has been removed and the fields rearranged a little. The use N_PortID was only required for proxy purposes. With the changes to that protocol, there is no further need for this addressing field.

We’ve removed from MANAGE ACL the ability to set the device back to the factory default unconstrained state. This is now only available in the DISABLE ACCESS CONTROLS service action.

PTPL (Persist Through Power-Loss) is now required. Our current feeling is that vendors will implement this feature in all cases, so having this optional only complicated the model unnecessarily.

2.5.4 Changes for revision 6

In this revision, the LUN Map owner switches from the target (access controls coordinator) to PAM. This affects the overall model in small ways, changes the parameter data for a number of service actions and more importantly, adds additional target requirements and service actions (mostly to deal with conflict resolutions issues).

Note that configuration conflicts may be detected by the access controls coordinator under two circumstances:

a) while processing a MANAGE ACL service action when the parameter data contains conflicting instructions; in this case, the target rejects the command;

b) when an initiator enrolls an AccessID whose ACL LUN Map instructions conflict with the ACL LUN Map instructions for that initiator's TransportID; in this case, we are suggesting that the target recover as best as possible (namely instantiate all non-conflicting parts of the LUN Map and log all conflicting parts).

We've changed the target-reset/power-cycle volatility of Proxy Tokens. In this revision, the Proxy Tokens are required to persist through these events, but any LUN Map entries created by ASSIGN PROXY LUN service actions are not (this is open for discussion however).

We've also documented in one place (2.1.5) all the open questions (that we are aware of).

2.5.5 Changes for revision 7

Closed on most of the open issues in section 2.1.3 of rev 6.

Changed the notion of iLUN to "default LUN", the LUN value used to address a logical unit in the absence of the access controls coordinator generating LUN Maps (e.g., the LUN values as seen in the default state or as would be "mapped" under the GrantAll page for MANAGE ACL).

Modified the REPORT LU DESCRIPTORS (formerly called DESCRIPTIONS) parameter data to include optional INQUIRY EVPD Page 83 identifier, change iLUN to "default LUN", added a generation value to the header, defined the additional data field. Modified the description of the parameter data used to indicate the range of LUN values the device has the ability to map.

Added generation value (for default LUN identifiers) to the MANAGE ACL service action.

Formalized on Proxy Tokens preserving through resets but that LUN Map entries created by ASSIGN PROXY LUN disappear through a reset.

Added a section (Appendix D.3) for requesting a bit in Standard INQUIRY data to indicate presence of an access controls coordinator.

Removed the REPORT LUN MAP service action. This contained no useful data (and to some folks, data that should not be made available to a generic initiator, namely the iLUN).

Fixed the length of all currently defined initiator identifiers (TransportID, both SPI and FCP, and AccessID) to 24 bytes, but left the extensibility of the MANAGE ACL and REPORT ACL parameter data structure with variable lengths.

Modified the specifics for ASC/ASCQ assignments per the request from Compaq (Rob Elliot). We have added additional ASC/ASCQ sense codes for OVERRIDE LOCKOUT and for LUN MAPPING CONFLICT (see 2.4 and Table 40).

Removed the VS bit and related facility from the DISABLE ACCESS CONTROLS service action. This now is defined to only reset the access controls to the default state when supplied with the correct key. Vendors who want to implement alternatives may use a VS-specific service action. This change was made because we feel that the model we've proposed for override of lost Management Identifier Key is sufficient to address this issue.

The current host/target/PAM interlock for LUN Map changes been accepted (by the author) as sufficient.

Defined a new method and model for override of lost keys (i.e., if PAM forgets her Management Identifier for a target).

Expanded the log concept and function to include other events (namely, key override events and invalid key events).

Added a section to the model clause to better encapsulate the resource requirements for access controls.

2.5.6 Changes for revision 8

Many editorial changes were made in this revision, primarily based on comments from Robert Elliot (numerous but minor), Ralph Weber (numerous and mostly major), and Charles Binford (thankfully, only a few). In particular, the summary table of service actions and the table of ASC/ASCQs has been moved out of the model clause. The service action table has additionally been split and included in each command subclause. A number of subclauses have been moved around to aid readability. We've ISO-fied the clause numbering and attempted to de-"which" the subordinate sentence clauses.

The ASC/ASCQ for ACCESS DENIED - OVERRIDE LOCKOUT has been removed; it wasn't needed.

Proposed glossary changes have been better formalized. Many items were removed; the model clause now has the official (and only) definition. Remaining glossary entries have added cross references to specific clauses.

We changed the Default LUN Generation value from "managed in a vendor-specific way" to "increased by one". Because of this, it is now required to be persistent.

The model clause has been significantly rewritten, both for clarity and for correctness/consistency. Most of the command/service action clauses have been modified, but typically in only minor ways, with the following exceptions:

a) Moved the CLEAR ACCESS CONTROLS LOG to the ACCESS CONTROL OUT command, putting the Log Portion field and Management Identifier Key into parameter data.
b) Split the previous function of the MANAGE OVERRIDE LOCKOUT TIMER of the ACCESS CONTROL IN command into two functions: REPORT OVERRIDE LOCKOUT TIMER of the ACCESS CONTROL IN and MANAGE OVERRIDE LOCKOUT TIMER of the ACCESS CONTROL OUT.

The Allocation Length requirements (e.g., "at least eight" else CHECK CONDITION) have been removed. The language now allows for any length. The Parameter List Length conditions have been cleaned up. A length of zero is always valid and always returns GOOD status, though no other action is taken.

Process Associators have been removed from the TransportID for FCP.

References to the term LUN Map have been removed. This avoids the problem of having to specify resources on how a LUN Map should be implemented. "LUN Mapping Conflicts" are now called "ACL LUN Conflicts". Any references to changes in an initiator's LUN Map have been changed to either changes in ACL entry or changes to an initiator's access rights, or related expressions.

In response to a AC OUT/MANAGE ACL service action that fails because there was a requested LUN value that is not supported by the access controls coordinator, the sense data is modified to include pointers to the invalid byte and the extended sense data contains an alternative LUN value.

Added a term called "access identifier" to be used in ACL entries. This term encompasses AccessIDs and TransportIDs. Along with this, the parameter data field name INITIATOR IDENTIFIER changed to ACCESS IDENTIFIER.

An ACL LUN Conflict at AC OUT/ACCESS ID ENROLL now fails completely and the initiator stays in the not-enrolled state. Previously, the command would succeed with RECOVERED ERROR and the "good" part of the ACL entry instantiated.

An AC OUT/ACCESS ID ENROLL when access controls are disabled now returns GOOD status, even though the initiator is not placed in the enrolled state. This is more consistent with other commands which success unconditionally if access controls are disabled and more correctly informative since the initiator does have access to logical units.

Changed the effects of an enrollment ACL LUN conflict because of proxy LUNs. Added this case to the ACL LUN conflict conditions and allowed for the same failure mechanism (without requiring a record in the access controls log).

Simplified the contents of the ACL LUN conflict log. It now only reports the TransportID and AccessID whose entries in the ACL caused the conflict at enrollment of the AccessID by an initiator with that TransportID.

Closed on the issues related to the Supported LUN-Mask format of REPORT LU DESCRIPTORS parameter data. No better alternative was proposed (yet?).

Changed the name of the INQUIRY IDENTIFICATION DESCRIPTOR field in REPORT LU DESCRIPTOR parameter data to EVPD IDENTIFICATION DESCRIPTOR.

Moved the specification of how access identifiers (AccessID in particular) are packed in parameter data to a separate major clause, recommending that it be added as a subclause of SPC-2 rev 16, clause 8, “Parameters for all device types”.

Changed the model clause formerly called “Establishment of access controls and other tasks” to “Interactions of Access Controls and other features” and changed the wording relating to AC changes and task states to the wording Ralph Weber proposed on the t10 reflector.

Reserved a few more of the service action codes (reduced the number of vendor-specific values). The AC OUT has grown to 10 actual service actions, and we were down to only 5 more reserved. We now have eight for vendor-specific (down from 16) and 24 for the standards to use. AC IN used 5 and AC OUT uses 10 of these 24. So there is still room to grow.

Changed the term “de-enrolled” to “pending-enrolled”. (Other reasonable choices were “semi-enrolled”, “weakly-enrolled” (math term, the author’s favorite), “partly-enrolled”, “incomplete-enrolled”, “partial-enrolled”, some combination of suspended and enrolled/enrollment.)

Modified most of the service actions to deal with the “access controls disabled” condition. In most cases, service actions return GOOD status (and in the case of AC IN service actions, return no data). In only a couple of Proxy-related service actions is the service action failed when access controls are disabled (see ASSIGN PROXY LUN and RELEASE PROXY LUN).

Modified the MANAGE ACL service action in the following two ways:

a) Page codes have simplified to two non-proxy page (and two proxy pages). The Grant and Revoke pages have been merged into one Grant/Revoke page. The Revoke All page has been deleted.

b) Prepended four Reserved bytes to the parameter data header. Now all OUT commands which require the Management Identifier Key have the key in bytes 4-11.

c) ACL entries are either added, removed or replaced in this model. In particular, parameter data in a MANAGE ACL service action does not update and existing entry, it replaces it.

d) Modified the rules for changing an initiator’s enrollment state to not-enrolled under MANAGE ACL service actions which affected its LUN map (when its ACL entry is replaced). In this version, a replacement forces the initiator into the not-enrolled state unless requested otherwise by PAM with a bit in parameter data.

2.5.7 Changes for revision 9

There are a handful of editorial changes to clarify wording and intent. The specific versions of the affected standards (SPC-3, SAM-2, FCP-2, SPI-4, SBC-2, MMC-3 and future RBC) have been included in this draft.

Additionally, we've rewritten the sections describing the changes for the other standards to be more specific about the required changes with recommendations for clause headings and insertion locations.

The only two technical issues are:

a) When an initiator in the enrolled state sends the ACCESS ID ENROLL service action with a different AccessID from the one under which it is currently enrolled, it now is transitioned to the pending-enrolled state.

b) Three service actions (REPORT ACL, REPORT LU DESCRIPTORS, REPORT ACCESS CONTROLS LOG) changed their behavior when access controls are disabled. In rev 8, these always returned status GOOD but no data. In this revision, they return only header (this includes the ADDITIONAL LENGTH field set to indicate that essentially no additional data is available).

3.0 Glossary and Acronyms

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

3.1 Glossary

Access Controls: An optional target 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 4.0.).

Access Control List: The data used by a target to configure access rights for initiators according to the access controls state of the device (see 4.1).

Access Controls Coordinator: The entity within a device that coordinates the management and enforcement of access controls (see 4.0) for all logical units within the device. This is always addressable through LUN 0.

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 4.0 and 4.6.2).

3.2 Acronyms

ACL: Access Control List (see 4.0)

4.0 Access Controls

AUTHOR'S NOTE: *This is the model clause for addition to SPC-3.*

4.1 Access Controls Overview

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

AUTHOR'S NOTE: *See Annexes A-E for the changes required in other standards documents.*

An application client may manage the access controls state of the target using access control commands:

a) ACCESS CONTROL IN - queries the access control information; and

b) ACCESS CONTROL OUT - creates, changes or revokes access controls, and otherwise manages the access controls coordinator.

The access control commands are not subject to reservation conflicts.

AUTHOR'S NOTE: *See Annex D for the changes required to Table 8 of SPC-2 (rev 14) with respect to reservation conflicts.*

The specific access controls of the device are instantiated by the access controls coordinator using data in an access controls list (ACL). The ACL consists of entries, each entry containing the following:

a) one **access identifier** (see 4.3);

b) a list of **accessible logical unit pairs**, each pair consisting of one LUN value and a reference to one logical unit.

Identification of logical units in an accessible logical unit pair is vendor specific. A logical unit shall be referenced in at most one accessible logical unit pair per ACL entry. A given LUN value shall appear in at most one accessible logical unit pair per ACL entry.

The contents of the ACL are managed by an application client via the ACCESS CONTROL OUT command with MANAGE ACL and DISABLE ACCESS CONTROLS service actions. Successful completion of these service actions require a Management Identifier Key value shared by the managing application client and the access controls coordinator (see 4.2, 6.2.2 and also 4.7). 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.

Every logical unit of a device shall be identified by a unique **default LUN value**. The default LUN value shall be the LUN value that would be reported in REPORTS LUNS 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.

The association of default LUN values and logical units is managed by the access controls coordinator and may change in ways beyond the scope of access controls. The access controls coordinator shall maintain a Default LUNs Generation value (see 4.1 and 4.2) that shall be used to time-stamp the association of default LUN values and logical units. This Default LUNs Generation value shall be increased by one each time the association of default LUNs to logical units changes.

> 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 first successful ACCESS CONTROL OUT command with MANAGE ACL service action enables access controls (see 6.2.2), 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 4.3) 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 4.6.2).

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 4.4). 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 4.1);
b) an 8-byte (64 bit) integer called the Management Identifier Key (see 4.1 and 6.2.2);
c) a 4-byte (32 bit) integer called the Default LUNs Generation (see 4.1);
d) a 2-byte (16 bit) integer called the Initial Override Lockout Timer (see 4.7);
e) a log of access controls related events containing at least the following (see 4.11):
    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 4.6.2).

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 4.8.

4.3 Access Identifiers

Initiators are identified in ACL entries on the basis of one or more of three types of access identifiers:

a) **AccessID**, as enrolled (see 4.4.1) by an initiator using the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 6.2.4);

b) **TransportID**, as defined in the relevant protocol or interconnect standard;

c) vendor-specific identifiers.

An AccessID shall be sixteen (16) bytes.

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.

Use of the TransportID is protocol and interconnect-specific. Each SCSI protocol standard may specify the description and use of the TransportID. A protocol specification for a TransportID shall not include address objects that do not persist across common reset events in the service delivery subsystem. Additionally, a TransportID shall be no more than twenty-four (24) bytes.

Access identifiers are included in parameter data as specified in 7.1.

4.4 Enrolling AccessIDs

4.4.1 Enrollment states

4.4.1.1 Summary of enrollment states

Initiators may enroll an AccessID with an access controls coordinator in order to gain access to logical units accessible via such an access identifier. An initiator shall be in one of three states with respect to such an enrollment:

a) **not-enrolled**: the state for an initiator when it first accesses the device and also the state entered into by the initiator in response to an ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action (see 6.2.5);

b) **enrolled**: the state an initiator enters as a consequence of a successful ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 6.2.4);

c) **pending-enrolled**: the state an initiator enters from the enrolled state as a consequence of certain events in the service delivery subsystem or by successful ACCESS CONTROL OUT commands with MANAGE ACL service action and FLUSH bit set to one (see 6.2.2).

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

4.4.1.2 Not-enrolled state

An initiator enters the not-enrolled state when it first accesses the device (sends any SCSI command). An initiator stays in this state until it successfully completes the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action. See 4.4.1.3 and 6.2.4.

An initiator in the enrolled or pending-enrolled state shall transition to the not-enrolled state as follows:

a) by successful completion of the ACCESS CONTROL OUT command with CANCEL ENROLLMENT service action (see 6.2.5);

b) as a consequence of power-cycles or target resets in a vendor-specific manner (see 4.8);

c) as a consequence of a successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action (see 6.2.2) that replaced an ACL entry for the enrolled AccessID if the NOCNCL bit is set to zero and, vendor-specifically, if the NOCNCL bit is set to one (see 6.2.2.2.2).

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 (e.g., MANAGE ACL service action or 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 is to provide an indication to the initiator that its access rights have changed, and consequently that its LUN addressing of logical units may have changed, by events or actions not taken by that initiator directly. 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 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. If the managing application client determines that these recovery actions are not required, the application client should set the NOCNCL bit to one to recommend to the access controls coordinator that it leave the initiator in its current enrollment state. A vendor has at least three implementation options for 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 the initiator (this may disrupt an initiator unnecessarily, but requires no extra resources on the part of the access controls coordinator).
c) Ignore the recommendation and instead examine the current and new access rights and LUN addressing to (independent of the managing application client) determine if the initiator should be transitioned. 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.

4.4.1.3 Enrolled state

The initiator enters the enrolled state from either the not-enrolled or pending-enrolled state by successful completion of the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action (see 6.2.4). This service action is successful only under the following conditions:

a) if the initiator was in the not-enrolled state and the AccessID in the parameter data of the service action matches the access identifier in an entry in the ACL (so that this AccessID has rights to one or more logical units); in this case, the initiator gains access to those logical units specified in the access grant logical unit pairs of that ACL entry;
b) if the initiator was in the enrolled or pending-enrolled state and the AccessID in the parameter data matches that of the current enrolled AccessID for that 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 parameter data does not match that of the current enrolled AccessID for that initiator, the device server shall respond with

CHECK CONDITION as specified in 6.2.4. 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 4.8.

Transitions out of the enrolled state are described in the subclauses for the not-enrolled (4.4.1.2) and pending-enrolled (4.4.1.4)states.

> NOTE 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 4.4.2.

4.4.1.4 Pending-enrolled state

An initiator shall enter the pending-enrolled state only from the enrolled state, and as a consequence of 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 in FCP or a hard bus reset for parallel SCSI);
b) successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action and FLUSH bit set to one;
c) optionally after a Target Reset task management function, as described in 4.8.

While in the pending-enrolled state, the initiator's access to logical units is limited according to the rules of 4.10.

4.4.2 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 the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action:

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

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

a) the access controls coordinator shall fail the enrollment and leave the initiator in the not-enrolled state;
b) except in the last case, the access controls coordinator shall record the event in the access controls log as described in 4.11;
c) the device server shall return status and sense data indicating that a conflict arose and that the enrollment failed (see 6.2.4).

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.

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 4.1 and 6.2.2). One ACL entry describes the access allowed to one access identifier (see 4.3), 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 4.6.2), 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 4.5 and 4.4.1.2 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 5.2.6). 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 D.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 6.2.11).

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 6.2.9 and 6.2.10);
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 6.2.2.2.3).

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 6.2.12);
b) an event in the service delivery subsystem causes the access controls coordinator to question whether the third party initiator that created the 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.

4.7 Override of Management Identifier Key

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 4 and Table 24 for a summary of the service actions requiring the Management Identifier Key.

However, conditions may arise when this key needs to be replaced and the current key is not available. In this case, the ACCESS CONTROL OUT command with OVERRIDE MGMT ID KEY service action may be used (see 6.2.8). This service action is intended only for failure scenarios; the MANAGE ACL service action should be used in all other circumstances.

To facilitate protection of the Management Identifier Key, the access controls coordinator shall support the following.

The access controls coordinator shall maintain a 16 bit non-negative integer-valued timer, called the Override Lockout Timer. This timer, if non-zero, 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 6.2.7). This service action has two functions, depending on whether or not the correct Management Identifier Key is supplied in the parameter data.

a) If the Management Identifier Key supplied in the parameter data is incorrect (or no parameter data is sent), the access controls coordinator shall restart the Override Lockout Timer, that is, reset it to its current initial value.

b) If the Management Identifier Key supplied in the parameter data is correct, then the access controls coordinator shall do the following:

   1.reset the Initial Override Lockout Timer value according to data in the parameter data;

   2.restart the timer to the new initial value.

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

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.

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.

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

The access controls coordinator is required to maintain in non-volatile form the entire access controls data as described in 4.2, including the access controls log (see 4.11).

If the device's non-volatile memory is not ready (to read the access controls data), the device server shall return on all addressed logical units a CHECK CONDITION status, a sense key set to NOT READY and additional sense data as defined in the TEST UNIT READY command (see SPC-2, rev 16, 7.27) for all commands except INQUIRY.

Additionally, all valid Proxy Tokens created as a consequence of ACCESS CONTROL IN commands with REQUEST PROXY TOKEN service action (see 5.2.6) 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 6.2.11) 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.

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 4.1 and 4.2). 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 4.11).

The REPORT OVERRIDE LOCKOUT TIMER service actions reports on the state of the Override Lockout Timer (see 4.7).

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 4.4.1.3) 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, 7.21.

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 4.4.2).

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 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 5.2.4.2.)

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 5.2.4.3.)

The ACL LUN conflict events occur as specified in 4.4.2. 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 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 5.2.4.4.)

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 5.2.4). 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 6.2.6).

5.0 ACCESS CONTROL IN command

5.1 ACCESS CONTROL IN command descriptor block

The ACCESS CONTROL IN command (see Table 3) is used to obtain information about the access controls that are active within the access controls coordinator and to facilitate other functions. The command shall be used in conjunction with the ACCESS CONTROL OUT command. It shall not be affected by reservations, persistent reservations or access controls.

If the device contains an access controls coordinator, this command shall be processed by the access controls coordinator if addressed to LUN 0 or to any other LUN value whose standard INQUIRY data has the ACC bit set to one. In the latter case, the command shall be processed in the same manner as if the command had been addressed to LUN 0. It shall be rejected by the device server if addressed to any other LUN with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to INVALID OPCODE.

TABLE 3. ACCESS CONTROL IN command

Byte	Bit 7	6	5	4	3	2	1	0
0	OPERATION CODE (86h)							
1	RESERVED			SERVICE ACTION				
2	MSB							
9		SERVICE ACTION-SPECIFIC DATA						LSB
10	MSB	SERVICE ACTION-SPECIFIC DATA2 or						
13		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 4.2.5 (of SPC-2 revision 16).

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 4 gives a summary of the ACCESS CONTROL IN command service action codes.

TABLE 4. ACCESS CONTROL IN command service action codes
(M=Mandatory, O=Optional, V=Vendor-specific)

Code	Name	Type	KeyRq	Clause
00h	REPORT ACL	M	Y	5.2.2
01h	REPORT LU DESCRIPTORS	M	Y	5.2.3
02h	REPORT ACCESS CONTROLS LOG	M	Y	5.2.4
03h	REPORT OVERRIDE LOCKOUT TIMER	M	Y	5.2.5
04h	REQUEST PROXY TOKEN	O	N	5.2.6
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.

5.2.2 REPORT ACL service action (Mandatory)

5.2.2.1 REPORT ACL service action command descriptor block

The REPORT ACL service action of the ACCESS CONTROL IN command is used by an application client to query the complete ACL currently maintained by the access controls coordinator.

The ALLOCATION LENGTH field should be at least eight (8), sufficient for the header information.

If access controls are disabled, the device server shall respond with GOOD status and return only the eight (8) byte header as specified in 5.2.2.2.1 subject to the ALLOCATION LENGTH limitation, regardless of the value of any other field in the CDB.

If access controls are enabled, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain the current Management Identifier Key maintained by the access controls coordinator. If this is not the case, the device server shall return no data and respond with CHECK CONDITION status, sense key set to 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 4.11).

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 5.2.2.2.

5.2.2.2 REPORT ACL parameter data format

5.2.2.2.1 REPORT ACL parameter data header

The format of the parameter data provided in response to an ACCESS CONTROL IN command with REPORT ACL service actions is shown in Table 5. The ACL Entry Page(s) are described in 5.2.2.2.2 and 5.2.2.2.3.

TABLE 5. REPORT ACL parameter data format

Byte	Bit 7	6	5	4	3	2	1	0
0	MSB							
3		ADDITIONAL LENGTH (*n*-3)						LSB
4	MSB							
7		DEFAULT LUNS GENERATION						LSB
8								
n	ACL Entry 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 in the requesting CDB. If access controls are disabled, the ADDITIONAL LENGTH field in the returned parameter data shall be set to four (4).

The DEFAULT LUNS GENERATION field shall be set to the current value of the Default LUNs Generation integer maintained by the access controls coordinator according to the rules in 4.1.

The ACL Entry Page(s) shall contain a description of the ACL maintained by the access controls coordinator. Each ACL Entry Page is identified by a Page Code. The list of Page Codes and their definitions is given in Table 6 and the detailed description of the pages are in subsequent subclauses.

TABLE 6. ACL Entry PAGE CODE definitions for REPORT ACL service action

Page Code	Description	Clause
00h	Granted	5.2.2.2.2
01h	Granted All	5.2.2.2.2
02h	Proxy Tokens	5.2.2.2.3
03h-FFh	Reserved	

5.2.2.2.2 REPORT ACL parameter data Granted and Granted All page formats

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 7. ACL Entry Page: Granted and Granted Default page formats

	Bit							
Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (00h-01h)							
1	RESERVED							
2	MSB							
3	PAGE LENGTH (*m*-3)							LSB
4	RESERVED							
5	IDENTIFIER TYPE							
6	MSB							
7	IDENTIFIER LENGTH (*n*-7)							LSB
8	MSB							
n	ACCESS IDENTIFIER							LSB
n+1								
m	LUN/DEFAULT LUN LIST							

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.

> NOTE All currently defined Identifier Types require the IDENTIFIER LENGTH field be set to 24 (see Table 33).

The LUN/DEFAULT LUN LIST field shall contain a list of LUN/default LUN pairs as specified in Table 8 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 8. LUN/DEFAULT LUN LIST format

Byte	Bit 7	6	5	4	3	2	1	0
0	MSB							
7		FIRST LUN						LSB
8	MSB							
15		FIRST DEFAULT LUN						LSB
		.						
		.						
		.						
n-15	MSB							
n-8		LAST LUN						LSB
n-7	MSB							
n		LAST DEFAULT LUN						LSB

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.

5.2.2.2.3 REPORT ACL parameter data Proxy Tokens page format

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

TABLE 9. ACL Entry Page: Proxy Tokens page format

Byte	Bit 7	6	5	4	3	2	1	0
0	PAGE CODE (02h)							
1	RESERVED							
2								
3	PAGE LENGTH (*m*-3)							
4								
m	PROXY TOKEN/DEFAULT LUN LIST							

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 with an empty PROXY TOKEN/DEFAULT LUN LIST field.

At most one Proxy Token page shall be included in the parameter data.

The PROXY TOKEN/DEFAULT LUN LIST field shall contain a list of Proxy Token/default LUN pairs as specified in Table 10 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 10. PROXY TOKEN/DEFAULT LUN LIST format

	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	MSB							
7		FIRST PROXY TOKEN						LSB
8	MSB							
15		FIRST DEFAULT LUN						LSB
		.						
		.						
		.						
n-15	MSB							
n-8		LAST PROXY TOKEN						LSB
n-7	MSB							
n		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.

5.2.3 REPORT LU DESCRIPTORS service action (Mandatory)

5.2.3.1 REPORT LU DESCRIPTORS command descriptor block

The REPORT LU DESCRIPTORS service action of the ACCESS CONTROL IN command is used by an application client to obtain from the access controls coordinator inventory information about the logical units for which access controls may be established and other properties of the access controls coordinator.

The ALLOCATION LENGTH field should be at least twenty (20), sufficient for the header information.

If access controls are disabled, the device server shall respond with GOOD status and return only the twenty (20) byte header as specified in 5.2.2.2.1 subject to the ALLOCATION LENGTH limitation, regardless of the value of any other field in the CDB.

> NOTE In this case, all logical units are accessible to all initiators; existing commands such as INQUIRY, REPORT LUNS, READ CAPACITY, etc., may be used to collect this information if needed.

If access controls are enabled, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain the current Management Identifier Key maintained by the access controls coordinator. If this is not the case, the device server shall return no data and respond with CHECK CONDITION status, sense key set to 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 4.11).

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 5.2.3.2.

5.2.3.2 REPORT LU DESCRIPTORS parameter data format

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 11.

TABLE 11. REPORT LU DESCRIPTORS parameter data format

Byte	Bit 7	6	5	4	3	2	1	0
0	MSB							
3		ADDITIONAL LENGTH (*n*-3)						LSB
4	MSB							
7		NUMBER OF LOGICAL UNITS						LSB
8								
15	SUPPORTED LUN-MASK FORMAT							
16	MSB							
19		DEFAULT LUNS GENERATION						LSB
20								
n	LOGICAL UNIT DESCRIPTORS							

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, the ADDITIONAL 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 shall be the same as the number of LOGICAL UNIT DESCRIPTORS that follow in the remaining parameter data).

The SUPPORTED LUN-MASK FORMAT field contains a summary of the LUN values that the access controls coordinator supports in an accessible logical unit pair in an ACL entry. The format is specified in Table 12.

TABLE 12. 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.

For example, if the access controls coordinator uses a flat addressing model and only supports LUN values at the top level and up to 256 LUNs, then the 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.

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 SUPPORT LUN-MASK FORMAT is intended only as a summary of the supported LUN values and not a complete description. 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 DEFAULT LUNS GENERATION field shall be set to the current value of the Default LUNs Generation integer maintained by the access controls coordinator according to the rules in 4.1.

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 13.

TABLE 13. LOGICAL UNIT DESCRIPTOR data format

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

The PERIPHERAL DEVICE-TYPE field shall be set according to the device type of the referenced logical unit as specified in Table 54 (of SPC-2, rev 16).

The ADDITIONAL 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.

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 field shall be supported if the device supports the INQUIRY command with EVPD bit set to one and Page Code set to 83h (Device Identification Page) and at least one identification descriptor has Association value of 0h (as defined in SPC-2 rev 16 8.4.3). In this case, the EVPD IDENTIFICATION DESCRIPTOR field shall be derived from one of these identification descriptors as follows:

a) if the identification descriptor has length less than or equal to thirty-two (32) bytes, then the EVPD IDENTIFICATION DESCRIPTOR field shall be set to the value of the 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 descriptor;

b) if the identification descriptor has 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 descriptor; additionally, the EVPD IDENTIFICATION DESCRIPTOR LENGTH field shall be set to 32;

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 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-2, rev 16, 7.26). 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-2, rev 16, 7.20) as follows:

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;

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 descriptor; additionally, the DEVICE IDENTIFIER LENGTH field shall be set to 32.

If no such 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.*

The DEVICE-TYPE SPECIFIC ADDITIONAL DATA field shall not be included unless otherwise specified in the device-type specific command set standard.

AUTHOR'S NOTE: *Appendix E defines the format for this field for device-types under SBC-2, MMC-3 and a future version of RBC. All other device-types do not seem to require any additional data of this type (at least, that's the author's current point of view).*

5.2.4 REPORT ACCESS CONTROLS LOG (Mandatory)

5.2.4.1 REPORT ACCESS CONTROLS LOG command descriptor block

The REPORT ACCESS CONTROLS LOG service action of the ACCESS CONTROL IN command is used by an application client to obtain from the access controls coordinator information from the access controls log (see 4.11).

The SERVICE ACTION-SPECIFIC DATA2 field in the CDB shall have the structure specified in Table 14.

TABLE 14. 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							
3	ALLOCATION LENGTH							LSB

The LOG PORTION field indicates to which portion of the log this service action applies, as specified in Table 15.

TABLE 15. LOG PORTION field definitions for REPORT and CLEAR ACCESS CONTROLS LOG service actions

LOG PORTION	Description	Clause
00b	key overrides	5.2.4.2
01b	invalid keys	5.2.4.3
10b	ACL LUN conflicts	5.2.4.4
11b	Reserved	

The ALLOCATION LENGTH field 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 5.2.4.2, 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 4.11);

b) otherwise, the device server shall return in parameter data that portion of the log indicated in the LOG PORTION field, as specified in 5.2.4.3 and 5.2.4.4.

For any value of the LOG PORTION field, if the access controls coordinator only supports the relevant event counter in the log and not the additional information, then the returned parameter data shall only contain the header information.

5.2.4.2 REPORT ACCESS CONTROLS LOG parameter data format for Key Overrides

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 16.

TABLE 16. REPORT ACCESS CONTROLS LOG parameter data format for key overrides

Byte	Bit 7	6	5	4	3	2	1	0
0	MSB							
3	ADDITIONAL LENGTH (*n*-3)							LSB
4	RESERVED							
5	RESERVED						LOG PORTION	
6	MSB							
7	KEY OVERRIDES COUNTER							LSB
8								
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 4.11). The format for these pages is found in Table 17.

TABLE 17. Key Overrides Log Page(s) data format

Byte	Bit 7	6	5	4	3	2	1	0
0								
2	RESERVED							
3	RESERVED							SUCCESS
4	MSB							
7	TIME STAMP							LSB
8	MSB							
31	TRANSPORTID							LSB
32	MSB							
33	INITIAL OVERRIDE LOCKOUT TIMER							LSB
34	MSB							
35	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 log was successful. A value of zero indicates that the command did not succeed.

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

The INITIAL OVERRIDE LOCKOUT TIMER field, 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 4.11.

5.2.4.3 REPORT ACCESS CONTROLS LOG parameter data format for Invalid Keys

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 18.

TABLE 18. REPORT ACCESS CONTROLS LOG parameter data format for invalid keys

Byte	Bit 7	6	5	4	3	2	1	0
0	MSB							
3	ADDITIONAL LENGTH (*n*-3)							LSB
4	RESERVED							
5	RESERVED						LOG PORTION	
6	MSB							
7	INVALID KEYS COUNTER							LSB
12								
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 4.11). The format for these entries is found in Table 19.

TABLE 19. Invalid Keys Log Page(s) data format

Byte	Bit 7	6	5	4	3	2	1	0
0								
1	RESERVED							
2	OPCODE							
3	RESERVED			SERVICE ACTION				
4	MSB							
7	TIME STAMP							LSB
8	MSB							
31	TRANSPORTID							LSB
32	MSB							
39	INVALID KEY							LSB

The OPCODE and SERVICE ACTION fields shall be set to the respective values from the CDB of the access controls command that contained the invalid key (in either the CDB or the associated parameter data).

The TIME STAMP field may be set to the value of the time stamp at the time the invalid key event was recorded. See 4.11.

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

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

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 20.

TABLE 20. REPORT ACCESS CONTROLS LOG parameter data format for ACL LUN conflicts

Byte	Bit							
	7	6	5	4	3	2	1	0
0	MSB							
3	ADDITIONAL LENGTH (*n*-3)							LSB
4	RESERVED							
5	RESERVED						LOG PORTION	
6	MSB							
7	ACL LUN CONFLICTS COUNTER							LSB
8								
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 4.11). The format for these entries is found in Table 21.

TABLE 21. ACL LUN Conflicts Log Page(s) data format

Byte	Bit 7	6	5	4	3	2	1	0
0 3	RESERVED							
4	MSB							
7	TIME STAMP							LSB
8	MSB							
31	TRANSPORTID							LSB
32	MSB							
55	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 4.11.

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.

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.

5.2.5 REPORT OVERRIDE LOCKOUT TIMER service action (Mandatory)

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 4.2 and 4.7).

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.

The ALLOCATION LENGTH field in the CDB should be at least eight (8), sufficient for the parameter data.

If access controls are enabled, the SERVICE ACTION-SPECIFIC DATA field in the CDB shall contain the current Management Identifier Key maintained by the access controls coordinator. If this is not the case, 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 4.11).

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 22.

TABLE 22. MANAGE OVERRIDE LOCKOUT TIMER parameter data format

Byte	Bit							
	7	6	5	4	3	2	1	0
0								
1	RESERVED							
2	MSB							
3	CURRENT OVERRIDE LOCKOUT TIMER							LSB
4	MSB							
5	INITIAL OVERRIDE LOCKOUT TIMER							LSB
6	MSB							
7	KEY OVERRIDES COUNTER							LSB

The CURRENT OVERRIDE LOCKOUT TIMER field shall be set to the current value of the Override Lockout Timer.

The INITIAL OVERRIDE LOCKOUT TIMER field shall be set to the value of the Initial Override Lockout Timer as established by the last successful ACCESS CONTROL OUT command with MANAGE OVERRIDE LOCKOUT TIMER service action (see 6.2.7).

The KEY OVERRIDES COUNTER field shall be set to the value of the Key Overrides Counter in the access controls log (see 4.11).

5.2.6 REQUEST PROXY TOKEN service action (Optional)

The REQUEST PROXY TOKEN service action of the ACCESS CONTROL IN command is used by an initiator to obtain from the access controls coordinator a Proxy Token for a logical unit to which it has non-proxy access rights. It may use this Proxy Token to grant a third-party temporary access to a logical unit. 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 device server shall return CHECK CONDITION status, with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN 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.

> NOTE The indicated response when access controls are disabled is sufficient for the initiator to determine that access controls are disabled. In this state, all logical units are accessible and all initiators share the same LUN value for addressing (this LUN value is the default LUN value). 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 ALLOCATION LENGTH field in the CDB should be at least eight (8), sufficient for a valid Proxy Token.

The SERVICE ACTION-SPECIFIC DATA field shall contain the Logical Unit Number 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 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 enrolled, and the access controls coordinator has sufficient resources, then the device server shall

respond with GOOD status and return in the parameter data an eight (8) byte Proxy Token. This token (while valid, see 4.6.2) 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 does not correspond to an accessible logical unit as indicated above, then the following rules apply:

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;

b) if the Logical Unit Number corresponds to a logical unit accessible only through a proxy token, 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;

c) if the Logical Unit Number 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 device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code 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 device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to INSUFFICIENT ACCESS CONTROL RESOURCES.

6.0 ACCESS CONTROL OUT Command

6.1 ACCESS CONTROL OUT command descriptor block

The ACCESS CONTROL OUT command (see Table 23) is 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 the ACCESS CONTROL IN command. This command shall not be affected by reservations, persistent reservations or access controls.

If the device contains an access controls coordinator, this command shall be processed by the access controls coordinator if addressed to LUN 0 or to any other LUN value whose standard INQUIRY data has the ACC bit set to one. In the latter case, the command shall be processed in the same manner as if the command had been addressed to LUN 0. It shall be rejected by the device server if addressed to any other LUN with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to INVALID OPCODE.

TABLE 23. ACCESS CONTROL OUT command

	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	OPERATION CODE (87h)							
1	RESERVED			SERVICE ACTION				
2 9	RESERVED							
10	MSB							
13	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.

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 24 gives a list of the ACCESS CONTROL OUT command service action codes.

TABLE 24. ACCESS CONTROL OUT command service action codes
(M=Mandatory, O=Optional, V=Vendor-specific)

Code	Name	Type	KeyRq	Clause
00h	MANAGE ACL	M	Y	6.2.2
01h	DISABLE ACCESS CONTROLS	M	Y	6.2.3
02h	ACCESS ID ENROLL	M	N	6.2.4
03h	CANCEL ENROLLMENT	M	N	6.2.5
04h	CLEAR ACCESS CONTROLS LOG	M	Y	6.2.6
05h	MANAGE OVERRIDE LOCKOUT TIMER	M	Y	6.2.7
06h	OVERRIDE MGMT ID KEY	M	N	6.2.8
07h	REVOKE PROXY TOKEN	O	N	6.2.9
08h	REVOKE ALL PROXY TOKENS	O	N	6.2.10
09h	ASSIGN PROXY LUN	O	N	6.2.11
0Ah	RELEASE PROXY LUN	O	N	6.2.12
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.

6.2.2 MANAGE ACL service action (Mandatory)

6.2.2.1 MANAGE ACL command descriptor block

The MANAGE ACL 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 service action adds, changes or removes an entry or multiple entries in the access controls coordinator’s ACL. This service action is mandatory if the ACCESS CONTROL OUT command is supported.

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 shall respond with GOOD status.

If the value in the PARAMETER LIST LENGTH field is less than twenty (20) or results in truncation of any ACL Entry Page as specified in 6.2.2.2, then the device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to PARAMETER LIST LENGTH ERROR. Otherwise, the structure of the parameter data shall be as described in 6.2.2.2.

Any of the following conditions in the parameter header or any parameter page require the device server to respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST, and additional sense code 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.

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 6.2.2.2.2 and 6.2.2.2.3.

TABLE 25. MANAGE ACL parameter list format

Byte	Bit 7	6	5	4	3	2	1	0
0 3	RESERVED							
4 11	MSB MANAGEMENT IDENTIFIER KEY							LSB
12 19	MSB NEW MANAGEMENT IDENTIFIER KEY							LSB
20	RESERVED							
21	FLUSH	RESERVED						
22	RESERVED							
23	RESERVED							
24 27	MSB LUNs GENERATION							LSB
28 *n*	ACL Entry Pages(s)							

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 4.11) 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.

The FLUSH bit of one instructs the access controls coordinator to transition every initiator in the enrolled state into the pending-enrolled state.

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

The ACL Entry page(s) that may follow in the parameter list provide additional changes to the ACL.

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 command cannot be processed with GOOD status;

b) if the command results in GOOD status, the following shall be instantiated as a single indivisible event:

1.changes dictated in the fields in the header of the parameter list are processed;

2.changes dictated by ACL Entry Pages are processed;

3.multiple ACL Entry Pages are processed sequentially;

4.if an ACL Entry Page contains conflicting instructions, the last instruction within the page takes precedence;

5.if an AccessID's ACL entry is replaced, then the any initiator in the enrolled or pending-enrolled state under that AccessID shall be transitioned to the not-enrolled state (see 4.4.1.2), unless indicated otherwise by a NOCNCL bit value of one in the ACL Entry Page (see 6.2.2.2.2).

An ACL Entry Page contains conflicting instructions if either of the following occurs:

a) two LUN/default LUN pairs appear with the same LUN value and different default LUN values, or

b) two LUN/default LUN pairs appear 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 26. Details of the contents of each page are described in subsequent subclauses.

TABLE 26. ACL Entry PAGE CODE definitions

Page Code	Action	Clause
00h	Grant/Revoke	6.2.2.2.2
01h	Grant All	6.2.2.2.2
02h	Revoke Proxy Token	6.2.2.2.3
03h	Revoke All Proxy Tokens	6.2.2.2.3
04h-FFh	Reserved	

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 27. Grant/Revoke and Grant All page formats

	Bit							
Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (00h-01h)							
1	RESERVED							
2 3	PAGE LENGTH (*m*-3)							
4	NOCNCL	RESERVED						
5	IDENTIFIER TYPE							
6 7	IDENTIFIER LENGTH (*n*-7)							
8 *n*	MSB ACCESS IDENTIFIER							LSB
n+1 *m*	LUN/DEFAULT LUN LIST (Grant) or DEFAULT LUN LIST (Revoke)							

The IDENTIFIER TYPE and ACCESS IDENTIFIER fields are described in 7.1. The IDENTIFIER LENGTH field indicates the number of bytes following taken up by the ACCESS IDENTIFIER field.

NOTE All currently defined Identifier Types require the IDENTIFIER LENGTH field be set to 24 (see Table 33).

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 (eight (8) bytes for each component of the pair). 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 CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code 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 7.22.1 (of SPC-2, revision 16) 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.

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) 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 IDENTIFIER TYPE indicates type AccessID and an initiator 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 4.4.1.2):

a) If the ACL entry corresponding to that AccessID is removed as a consequence of the Grant/Revoke page, the initiator is transitioned to the not-enrolled state.
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 zero in that page, the initiator is transitioned to the not-enrolled state.
c) 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 page, then the initiator may be transitioned to the not-enrolled state in a vendor-specific manner.

6.2.2.2.3 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 28. 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 3	PAGE LENGTH (*m*-3)							
4 *m*	PROXY TOKEN LIST							

The PAGE LENGTH field shall indicate the number of additional bytes required for 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 access controls coordinator to revoke each of the listed Proxy Tokens. 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.

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 pages may be included in the parameter data. They are processed sequentially.

6.2.3 DISABLE ACCESS CONTROLS service action (Mandatory)

The DISABLE ACCESS CONTROLS service action of the ACCESS CONTROL OUT command is used by an application client to return the access controls coordinator to access controls disabled state.

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 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 CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is twelve (12), the parameter list shall be as described in Table 29.

TABLE 29. DISABLE ACCESS CONTROLS parameter list format

Byte	Bit							
	7	6	5	4	3	2	1	0
0 3	RESERVED							
4	MSB							
11	MANAGEMENT IDENTIFIER KEY							LSB

If access controls are enabled, the MANAGEMENT IDENTIFIER KEY field shall match the current Management Identifier Key maintained by the access controls coordinator. If this is not the case, the device server shall return CHECK CONDITION with sense key ILLEGAL REQUEST, additional sense data of 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 4.11) 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 disable access controls, clear the ACL, transition all initiators into the not-enrolled state, set the Management Identifier Key to zero, clear the access controls log (including resetting counters to zero) with the exception of the key overrides portion of the log (see 4.11), and allow all initiator's access to all logical units at their default LUN value. Optionally, the access controls coordinator may reset the Default LUNs Generation to zero (see 4.1).

6.2.4 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.

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 shall respond with GOOD status.

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 34.

If the value in the PARAMETER LIST LENGTH field is neither zero nor twenty-four (24), the device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to PARAMETER LIST LENGTH ERROR.

If the initiator is in the enrolled or pending-enrolled state under a given AccessID and the parameter data contains a different AccessID, then the device server shall respond with CHECK CONDITION status, with sense key set to ILLEGAL REQUEST and additional sense code 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 given AccessID and the parameter data 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 change to the access rights for that initiator.

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 4.4.2),

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 4.4.

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 4.4.2).

then the device server shall respond with 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.

> NOTE1: If an initiator receives ACL LUN CONFLICT sense 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. (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 CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to ACCESS DENIED - NO ACCESS RIGHTS.

6.2.5 CANCEL ENROLLMENT service action (Mandatory)

The CANCEL ENROLLMENT service action of the ACCESS CONTROL OUT command is used by an initiator to remove its enrollment with the access controls coordinator. Successful completion of this command changes the state of the initiator to the not-enrolled state.

This command should be used by an initiator prior to any period where use of its accessible logical units will be suspended for an extensive period of time (e.g., if the host is preparing to shutdown). This allows the access controls coordinator to free any resources allocated to manage the enrollment for that initiator.

There is no parameter data for this command. The PARAMETER LIST LENGTH field in the CDB for this service action shall be set to zero. If not, the device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to PARAMETER LIST LENGTH ERROR.

If the Parameter List Length field is set to zero, 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 4.10.

6.2.6 CLEAR ACCESS CONTROLS LOG service action (Mandatory)

The 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.

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 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 CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is twelve (12), the parameter list shall be as described in Table 31.

TABLE 30. CLEAR ACCESS CONTROLS LOG parameter list format

Byte	Bit							
	7	6	5	4	3	2	1	0
0								
1	RESERVED							
2	RESERVED							
3	RESERVED						LOG PORTION	
4	MSB							
11	MANAGEMENT IDENTIFIER KEY							LSB

The LOG PORTION field of this structure shall be interpreted according to Table 15.

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

If access controls are enabled, the MANAGEMENT IDENTIFIER KEY field shall match the current Management Identifier Key maintained by the access controls coordinator. If this is not the case, the device server shall return CHECK CONDITION with sense key ILLEGAL REQUEST, additional sense data of 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 4.11) 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):

1. preset the access controls log counter to zero;
2. clear the additional access controls log information.

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

6.2.7 MANAGE OVERRIDE LOCKOUT TIMER service action (Mandatory)

The 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 4.7).

If access controls are disabled, the device server shall respond with GOOD status, regardless of the value of any other field in the CDB.

If the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall restart the Override Lockout Timer (reset the value of this timer to the current Initial Override Lockout Timer) 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 CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to PARAMETER LIST LENGTH ERROR.

If the value in the PARAMETER LIST LENGTH field is twelve (12), the parameter list shall be as described in Table 31.

TABLE 31. MANAGE OVERRIDE LOCKOUT TIMER parameter list format

Byte	Bit							
	7	6	5	4	3	2	1	0
0								
1	RESERVED							
2	MSB							
3	NEW INITIAL OVERRIDE LOCKOUT TIMER							LSB
4	MSB							
11	MANAGEMENT IDENTIFIER KEY							LSB

If access controls are enabled, the MANAGEMENT IDENTIFIER KEY field shall match the current Management Identifier Key maintained by the access controls coordinator. If this is not the case, the device server shall return CHECK CONDITION with sense key ILLEGAL REQUEST, additional sense data of 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 4.11), 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:

1. preset the Initial Override Lockout Timer to the value of the NEW INITIAL OVERRIDE LOCKOUT TIMER field in parameter data;
2. reset the Override Lockout Timer to the new initial value.

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

6.2.8 OVERRIDE MGMT ID KEY service action (Mandatory)

The 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 maintained by the access controls coordinator. This is intended to be used in a failure situation where the managing application client no longer has access to its copy of this key.

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 shall respond with GOOD status.

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 4.11.

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

If the value in the PARAMETER LIST LENGTH field is twelve (12), the parameter data shall be as specified in Table 32.

TABLE 32. OVERRIDE MGMT ID KEY parameter list format

Byte	Bit							
	7	6	5	4	3	2	1	0
0 3	RESERVED							
4	MSB							
11	NEW MANAGEMENT IDENTIFIER KEY							LSB

The NEW MANAGEMENT IDENTIFIER KEY field shall contain a new Management Identifier Key.

If the Override Lockout Timer managed by the access controls coordinator is non-zero, then the device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST, and additional sense code 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 the current Management Identifier Key to the value in the parameter data. The device server shall respond with GOOD status.

6.2.9 REVOKE PROXY TOKEN service action (Optional)

The 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. 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 device server shall return CHECK CONDITION status, and sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.

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 shall respond with GOOD status.

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

If the 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.

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;

b) deny access to that logical unit by any initiator whose rights were granted under that Proxy Token by an ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action according to the rules of 4.10.

6.2.10 REVOKE ALL PROXY TOKENS service action (Optional)

The 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 logical unit that were granted to third parties under all

Proxy Tokens. 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 device server shall return CHECK CONDITION status, and sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.

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 shall respond with GOOD status.

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

If the LUN value is not associated to a logical unit to which the requesting initiator has any access rights, 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 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 referenced by the LUN value;

b) deny access to that logical unit by any initiator whose rights were granted under any Proxy Token by a ASSIGN PROXY LUN service action.

6.2.11 ASSIGN PROXY LUN service action (Optional)

The 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 and to assign that logical unit a particular LUN value for addressing by that 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 device server shall return CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN CDB.

If the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and the device server shall respond with GOOD status.

If the value in the PARAMETER LIST LENGTH field is neither zero nor sixteen (16), the device server shall respond with CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code 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.

If the Proxy Token is not valid, then the device server shall return CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to ACCESS DENIED - INVALID PROXY TOKEN.

NOTE If access controls are disabled, there can be no valid Proxy Tokens. Consequently, in this state, the device server always responds with the indicated error status. (This is different from many

other service actions where the response is GOOD status if access controls are disabled; it is used to inform the initiator that its request for the new LUN assignment failed.)

If the Proxy Token is valid but the access controls coordinator cannot assign the requested LUN value to the associated logical unit (either because the LUN value already is associated to a logical unit accessible to that initiator or because the LUN value cannot be supported as a valid logical unit address), then the device server shall return CHECK CONDITION status, sense key set to ILLEGAL REQUEST and additional sense code set to ACCESS DENIED - INVALID LU IDENTIFIER. Furthermore, the sense data shall be modified as follows. The SENSE-KEY SPECIFIC bit shall be set as described in 7.22.1 (of SPC-2 revision 15) 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 next eight bytes (if available) beyond the last byte of the FIELD POINTER field may include a LUN value that the access controls coordinator could support for this proxy token. In this case, no new access rights are granted to the initiator.

If the Proxy is valid but the access controls coordinator has insufficient resources to perform the requested action, then the device server shall respond with CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INSUFFICIENT ACCESS CONTROL RESOURCES.

If the Proxy is valid and the access controls coordinator has sufficient resources, then the device server shall return GOOD status and allow proxy access for that initiator to the referenced logical unit at that LUN address.

6.2.12 RELEASE PROXY LUN service action (Optional)

The RELEASE PROXY LUN service action of the ACCESS CONTROL OUT command is used by an initiator to remove a proxy access right to a logical unit created with a Proxy Token and the ASSIGN PROXY LUN service action. 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 device server shall return CHECK CONDITION status, sense key set to ILLEGAL REQUEST, and additional sense code set to INVALID FIELD IN CDB.

This command should be used by an initiator when its access to that logical unit is no longer required under its proxy rights (e.g., when a copy server has completed a specific third party copy service under the proxy). This allows the access controls coordinator to free any resources allocated to manage the proxy for that initiator.

If the PARAMETER LIST LENGTH field in the CDB is zero, the access controls coordinator shall take no action and the device server shall respond with GOOD status.

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

If the LUN value was not assigned to a logical unit by an ASSIGN PROXY LUN service action, the device server shall return CHECK CONDITION status, with sense key set to ILLEGAL REQUEST and additional sense code set to INVALID FIELD IN PARAMETER LIST.

NOTE If access controls are disabled, there can be no valid Proxy Tokens and therefore no LUN value could be assigned to a logical unit by an ASSIGN PROXY LUN service action. Consequently, in this state, the device server always responds with the indicated error status. (This is different from many other service actions where the response is GOOD status if access controls are disabled and is used to inform the initiator 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 ASSIGN PROXY LUN service action, the access controls coordinator shall disallow access to the logical unit at this LUN address and the device server shall return GOOD status.

7.0 Access Controls parameters

7.1 Access identifiers

Access identifiers are used in conjuction with access controls (see 4.0 and specifically 4.3) 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 33. 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 format of the AccessID data structure is described in Table 34. There are sixteen (16) bytes of significant data in this structure.

TABLE 34. AccessID data structure

	Bit							
Byte	7	6	5	4	3	2	1	0
0	MSB							
15		ACCESSID						LSB
16								
23	RESERVED							

The structure of the parameter data for a TransportID is protocol and interconnect-specific and may be specified in the relevant SCSI protocol standard.

A Changes required in SAM-2.

AUTHOR'S NOTE: *The unprefixed Table and Section numbers reference tables and sections of this document and will need to be updated as this proposal is included in SPC-3 and SAM-2. The Table and Section numbers prefixed with "SAM-" in the descriptions below refer to SAM-2 (rev 13).*

This clause contains some changes required in SAM-2 to deal with Task Management in the presence of access control.

A.1 Changes for the end of clause SAM-6.0.

The device server response to task management requests is subject to the access control state of the access controls coordinator (as instantiated by ACCESS CONTROL OUT commands) as follows:

a) a task management request of ABORT TASK, ABORT TASK SET or CLEAR ACA shall be unaffected by the presence of access restrictions;

b) a task management request of CLEAR TASK SET or LOGICAL UNIT RESET received from an initiator that is denied access to the logical unit (either because it has no access rights or because it is in the pending-enrolled state) shall cause no change to the logical unit, but shall receive a response of FUNCTION COMPLETE.

c) a TARGET RESET task management request shall initiate a logical unit reset as described in SAM-5.6.7 for all logical units to which the initiator has access, and shall cause no change to any logical units to which the initiator is denied access. A response of FUNCTION COMPLETE shall be returned in the absence of any other error condition.

A.2 Additions for clause SAM-5.6.6

While the device server response to task management requests is subject to the access rights of the requesting initiator, a target hard reset in response to a reset event within the service delivery subsystem shall be unaffected by access control.

B Changes required in FCP-2.

AUTHOR'S NOTE: *It may be that some of this could be included within SPC-3. This needs to be addressed with the appropriate editors. Certainly, the text and table of B.1 might be more appropriately placed in SPC-3, particularly, if all of Appendix C ends up in SPC-3.*

AUTHOR'S NOTE: *The unprefixed Table and Section numbers reference tables and sections of this document and will need to be updated as this proposal is included in SPC-3 and FCP-2. The Table and Section numbers prefixed with "FCP-" in the descriptions below refer to FCP-2 (rev 04).*

This clause contains the changes required in FCP-2. This includes the description of the TransportID.

B.1 Specification of the TransportID for Access Controls

The following text should be inserted as a separate subclause (at the editor's discretion) within clause FCP-10 with section heading "Specification of the TransportID for Access Controls". An alternative is to place this clause after clause FCP-5.2.

SCSI access controls (as defined in SCSI Primary Commands-3) use access identifiers in parameter data for certain access control command service actions to identify one or more initiators for the purpose of granting or reporting on access rights to logical units. The protocol-specific access identifier is called the TransportID. When used in parameter data, the TransportID structure for the parallel interface is 24 bytes long and is described in Table 35.

TABLE 35. TransportID for FCP.

Byte	Bit 7	6	5	4	3	2	1	0
0	RESERVED						PN_VAL	NN_VAL
1 7	RESERVED							
8 15	MSB WWPORTNAME							LSB
16 23	MSB WWNODENAME							LSB

A PN_VAL bit of one indicates that the WWPORTNAME field is valid. Similarly, the NN_VAL bit of one indicate that the WWNODENAME field is valid. A value of zero for any of these bits indicate that the corresponding field is invalid and shall be ignored. At least one of these validity bits must be set to one. If not, then the TransportID is invalid.

If both WWN fields are valid but are inconsistent, that is, they do not correspond to a device in the fabric, then the TransportID is invalid.

B.2 Affect on enrollment state under PRLO

CHANGE in the first paragraph in FCP-6.3 from:

... after a SCSI hard reset or power on reset. If any image pairs between the initiator and the host remain after the PRLO, then there is no clearing effect on any task, reservation, mode page parameter or status.

TO:

... after a SCSI hard reset or power on reset. The access controls enrollment state (see SCSI Primary Commands-3) of the initiator is transitioned to the not-enrolled state. If any image pairs between the initiator and the host remain after the PRLO, then there is no clearing effect on any task, reservation, mode page parameter, status or access controls enrollment state.

B.3 Volatility of the AccessID enrollments:

TABLE 36. Changes to FCP-Table 4: Clearing effects of SCSI Initiator Actions

	POWER	RESETLIP	LOGO, PLOGI	ABTS	PRLI, PRLO	TPRLO	TGTRESET	CLEAR	ABORT	LURESET
Access controls data (see SPC-3, 4.2 and 4.8)	N	N	N	N	N	N	N	N	N	N
AccessID enrollment state to pending-enrolled state (see SPC-3, 4.4.1) For all SCSI initiators in enrolled state	Y[a]	Y	Y[b]	N	Y[b]	Y[c]	Y[a]	N	N	N
Only for SCSI initiator port initiating action in enrolled state	-	-	Y[6]	N	Y	-	N	N	N	N

a. Transition is to pending-enrolled or not-enrolled state in vendor-specific manner (see SPC-3, 4.8)

b. For PRLO only and for explicit or implicit LOGO only

c. Only for the initiator attached to the port in the third party logout page.

C Changes required in SPI-4.

AUTHOR'S NOTE: *It may be that all this could be included within SPC-3. This needs to be addressed with the appropriate editors.*

AUTHOR'S NOTE: *The unprefixed Table and Section numbers reference tables and sections of this document and will need to be updated as this proposal is included in SPC-3 and SPI-4. The Table and Section numbers prefixed with "SPI-" in the descriptions below refer to SPI-4 (rev 00).*

This clause contains the changes required in SPI-4. This includes the description of the TransportID.

The following clauses should be inserted as a separate subclause (at the editor's discretion) within clause SPI-18. References to SPC-3 may not need to be explicit by clause; they are included here for context.

18.2 SCSI Access Controls

18.2.1. Specification of the TransportID for Access Controls

SCSI access controls (as defined in SCSI Primary Commands-3) use access identifiers in parameter data for certain access control command service actions to identify one or more initiators for the purpose of granting or reporting on access rights to logical units. The protocol-specific access identifier is called the TransportID. When used in parameter data, the TransportID structure for the parallel interface is 24 bytes long and is described in Table 37.

TABLE 37. TransportID for SPI.

Byte	Bit							
	7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0								
1	RESERVED							
2	MSB							
3	SCSI ADDRESS							LSB
4								
23	RESERVED							

The SCSI ADDRESS field indicates the SCSI address of the initiator.

AUTHOR'S NOTE: *The SCSI Address is defined in the glossary of SPI-4 (rev 00) in item SPI-3.1.82.*

18.2.2. Volatility of the AccessID enrollments

The access controls enrollment state of an initiator (established initially with the ACCESS CONTROL OUT command with ACCESS ID ENROLL service action, see SCSI Primary Commands-3) shall be transitioned by the following events or states:

a) power cycle of the device server;

b) hard reset bus condition.

The resulting transition state is implementation-specific but is subject to the rules of access controls as specified in SCSI Primary Commands-3.

D Additional Changes to SPC-3

AUTHOR'S NOTE: *The unprefixed Table and Section numbers reference tables and sections of this document and will need to be updated as this proposal is included in SPC-3. The Table and Section numbers prefixed with "SPC-" in the descriptions below refer to SPC-2(rev 18).*

D.1 Changes to SPC-Table 8

The following additional line(s) need to be added SPC-Table 8.

TABLE 38. Additional rows for SPC-Table 8

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/OUT	Allowed	Allowed	Allowed	Allowed	Allowed	Allowed

D.2 Changes to EXTENDED COPY

In the target descriptor formats in SPC-Tables 24, 26, 27, 28, 29, and 30, change byte3, bits 0-1 to a new 2-bit field called LU ID TYPE. In SPC-Table 26, 27, 28, and 29, change the LOGICAL UNIT NUMBER field name to LU IDENTIFIER. Add the following paragraphs to clause SPC-7.5.6.1 after the paragraph that begins "The copy manager may,...":

The LU ID TYPE field determines the interpretation of the LU IDENTIFIER field in some target descriptors (see SPC-7.5.6.2, SPC-7.5.6.3, SPC-7.5.6.4, and SPC-7.5.6.5). This is described in Table 39. In all other target descriptors this field is reserved.

TABLE 39. 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 values other than 00b (Logical Unit Number) is optional. If a copy manager receives an unsupported LU ID Type code in a target descriptor, the command shall 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 PARAMETER LIST.

If the LU ID TYPE field indicates Logical Unit Number, then the LU IDENTIFIER field specifies the logical unit within the SCSI device 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 indicates Proxy Token, then the LU IDENTIFIER field specifies an access controls Proxy Token (see 4.6.2) that shall be used by the copy manager to gain proxy access rights to the relevant logical unit 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 ACCESS CONTROL OUT command with ASSIGN PROXY LUN service action (see 6.2.11) 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 the LUN assigned on the basis of this Proxy Token only those commands that are necessary for the completion of those EXTENDED COPY commands that contain this Proxy Token value in their target descriptors. When the copy manager has completed EXTENDED COPY commands involving that Proxy Token, the copy manager should release the LUN value by sending the ACCESS CONTROL OUT command with RELEASE PROXY LUN service action (see 6.2.12) to the access controls coordinator of the SCSI device identified in the target descriptor.

In each subclause SPC-7.5.6.2-7.5.6.5, 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-7.5.6.1.

In the subclause SPC-7.5.6.6, insert the following paragraph after the paragraph which starts “The contents of...”

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

D.3 Changes to Standard INQUIRY parameter data

In the standard INQUIRY data format, SPC-Table 53, 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-7.6.2:

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.

D.4 Changes to additional sense codes and additional sense code qualifiers tables

The contents of Table 40 should be merged with SPC-Table 115 and its companion SPC-Table C.1.

TABLE 40. ASC and ASCQ assignments relevant to Access Controls

ASC	ASCQ	D T L P W R S O M C A E B K	Description
20h	01h	D T L P W R S O M C A E B K	ACCESS DENIED - INITIATOR PENDING-ENROLLED
20h	02h	D T L P W R S O M C A E B K	ACCESS DENIED - NO ACCESS RIGHTS
20h	03h	D T L P W R S O M C A E B K	ACCESS DENIED - INVALID MGMT ID KEY
20h	04h	D T L P W R S O M C A E B K	ACCESS DENIED - ENROLLMENT CONFLICT
20h	05h	D T L P W R S O M C A E B K	ACCESS DENIED - INVALID LU IDENTIFIER
20h	06h	D T L P W R S O M C A E B K	ACCESS DENIED - INVALID PROXY TOKEN
20h	07h	D T L P W R S O M C A E B K	ACCESS DENIED - ACL LUN CONFLICT
55h	05h	D T L P W R S O M C A E B K	INSUFFICIENT ACCESS CONTROL RESOURCES

E Changes to SBC-2, MMC-3 and future RBC

AUTHOR'S NOTE: *It is possible that this information could be included in SPC-3, in the clause on LU DESCRIPTORS. This needs to be discussed with the appropriate editors.*

AUTHOR'S NOTE: *In all of these sections, we're assuming that the LONGLBA bit is defined and supported in the standard (see 99-259r4). Additionally, we leave to the discretion of the editor of the relevant documents the appropriate clause or subclause under which these changes should be placed though a recommendation is given below.*

AUTHOR'S NOTE: *The references to SPC-3 below will need to be verified after this proposal is included in that document, if specific clause references are required.*

We need to add an additional clause to each of these standards that describes the DEVICE-TYPE SPECIFIC ADDITIONAL DATA field in the REPORT LU DESCRIPTORS logical unit descriptor parameter data. This new clause should contain the following information.

E.1 Changes to SBC-2 for REPORT LU DESCRIPTORS service action

AUTHOR'S NOTE: *This clause may go after the clause in SBC-2 which describes Mode parameters within the clause for "Parameters for direct-access block devices", under a new clause with heading "Access Controls LU Descriptors".*

The DEVICE-TYPE SPECIFIC ADDITIONAL DATA field in the logical unit descriptor parameter data for the ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action (see SPC-3, 5.2.3) shall defined as follows.

For all Peripheral Device Types covered by this standard, if the RMB bit in the Standard INQUIRY data indicates non-removable medium (RMB equal zero), then this field shall be twelve (12) bytes long. The data shall match the data that would be returned for a successful READ CAPACITY command with LONGLBA bit set to one, and RELADR and PMI bits set to zero. If the RMB bit indicates removable medium (RMB equal to one), this field shall be empty (zero bytes long).

E.2 Changes to future revisions of RBC for REPORT LU DESCRIPTORS service action

AUTHOR'S NOTE: *This clause may be inserted within the clause in RBC-x which describes SPC-x Implementation Requirements for RBC devices as follows. Either add an additional clause after the "Mode parameters" clause with heading "Access Controls LU Descriptors" and the text given below. Or alternatively, (1) add a row to the Required SPC-x Commands Table (Table 15 of RBC rev10a) for ACCESS CONTROL IN, with "O" in the "Command Support" columns for both "Fixed" and "Removable" and (2) add an additional subclause with heading "ACCESS CONTROL IN command" with the text below.*

The DEVICE-TYPE SPECIFIC ADDITIONAL DATA field in the logical unit descriptor parameter data for the ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action (see SPC-3, 5.2.3) shall defined as follows.

For RBC devices, if the RMB bit in the Standard INQUIRY data indicates non-removable medium (RMB equal zero), then this field shall be twelve (12) bytes long. The data shall match the data that would be returned for a successful READ CAPACITY command with LONGLBA bit set to one, and RELADR and PMI bits set to zero. If the RMB bit indicates removable medium (RMB equal to one), this field shall be empty (zero bytes long).

E.3 Changes to MMC-3 for REPORT LU DESCRIPTORS service action

AUTHOR'S NOTE: *This clause may go after the clause in MMC-3 which describes Mode parameters within the clause for "Parameters for all Logical Unit types", under a new clause with heading "Access Controls LU Descriptors".*

The DEVICE-TYPE SPECIFIC ADDITIONAL DATA field in the logical unit descriptor parameter data for the ACCESS CONTROL IN command with REPORT LU DESCRIPTORS service action (see SPC-3, 5.2.3) shall defined as follows.

For all Peripheral Device Types covered by this standard, if the RMB bit in the Standard INQUIRY data indicates non-removable medium (RMB equal zero), then this field shall be twelve (12) bytes long. The data shall match the data that would be returned for a successful READ CAPACITY command with LONGLBA bit set to one, and RELADR and PMI bits set to zero. If the RMB bit indicates removable medium (RMB equal to one), this field shall be empty (zero bytes long).