Date:November 1, 1999

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 device server actions to implement access control management. Two new commands are proposed (DataIn and DataOut) which allow configuration (DataOut) and reporting (DataIn) of access control management functions at the device server. The new commands and actions are not restricted to storage devices but are applicable (or extendable) to any device server.

This draft reflects comments, questions and suggestions from folks at Adaptec, Compaq and others at IBM.

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:

1. they should be enforced at the device server;

2. they should be granted to a host (i.e., at the OS-level) and not to particular initiators (or HBAs) within a host;

3. they should be configured by some application client which is responsible for overseeing access controls over the entire SAN;

4. 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 inadaquate 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 can instruct a given target device to restrict access to itself by all initiators except those from some small set. Such a set might be a single initiator. 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”. Initiators outside this group are denied (most) access to the device. In particular, these initiators can not preempt a reservation, issue read/write commands and the like. (Note: 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 can be applicable to any kind of device server, not just storage devices. Resource requirements at the device server can 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 DataIn-type command to query various status information of the device server with respect to access management 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 can be instantiated by a real application or instantiated more generally by the use of the defined protocol by users.

DataOut-type command to configure different kinds of access restrictions. These are detailed in later sections.

Hosts (at the OS-level) can 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[1], 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 inadaquate. 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 and 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 on that host or any actions which change the hardware configuration of the machine. This makes management by PAM of the device server access controls much simpler. But it also implies requirements on the part of device server to maintain associations between the AccessID and a given initiator's port or ports. These requirements are similar to but in some cases less restrictive than those already required by reservations.

AUTHOR'S NOTE: *Modifications to this description would be required in the context of Fibre Channel Process Associators.*

What follows is a detailed description of the new commands and device server requirements and constitutes the normative part of the proposal.

1.A host machine might refer to a virtual machine running within the context of other virtual machines on the same hardware.

2.0 Access Controls

Access controls may be used to limit the set of initiators which can execute certain commands at a device server. The device server shall reject certain commands from initiators outside the specified set. Initiators can be identified uniquely by an access identifier, called an AccessID, as defined in this proposal or by transport-layer identifiers defined in the relevant transport-protocol addendums.

An application client may add or remove initiators from the selected set using access control commands.

Unlike reservations, there are only two types of access rights for an initiator:

5. unrestricted access - all commands are handled in their normal fashion, and
6. restricted access - only certain commands are accepted and access-restricted commands are rejected with an error condition.

The access control commands are not subject to reservation conflicts though some service actions (e.g., PROXY ACCESS) are subject to access controls..

The scope of an access control shall be one of the following:

1. **logical unit** - a logical unit access control restricts access to the entire logical unit; and
2. **element -** an element access control restricts access to a specified element within a medium changer.

The methods of managing access controls are identified by the new commands:

1. ACCESS CONTROL IN - used to query the access controls; and
2. ACCESS CONTROL OUT - used to create, change or revoke access controls.

The default state of the device is to allow unrestricted access to the device by all initiators (subject to the existing protocols). That is, access controls are initially disabled for all scopes defined at the device server. This remains the case until the first successful completion of an ACCESS CONTROL OUT command with MANAGE ACL service action. This same service action can selectively disable or reenable access controls for any specific scope.

If a device server supports the access control commands, it shall be able to maintain at least one entry in its access control table for each logical unit. In this way, each logical unit can be dedicated to at least one initiator and so restrict access to competing initiators.

Along with the access control list, the device server should maintain an Access Controls Generation value of 8 bytes (64 bit integer). The default state for this value is zero. The value is managed by ACCESS CONTROL OUT commands with MANAGE ACL service actions.

For each command, this standard or a related command standard for the particular device type defines the conditions under which the command from a particular initiator is or is not subject to access controls. In general, any command which is not restricted by reservations of any type is not restricted by access controls. The SPC-commands not subject to access controls are:

1. INQUIRY;
2. LOG SENSE;
3. PREVENT/ALLOW, PREVENT equal zero;
4. REPORT DEVICE IDENTIFIER;
5. REPORT LUNS;
6. REQUEST SENSE.

AUTHOR'S NOTE: *I would include MODE SENSE in the above list as it only requests information about the device and not the user data. It is left out of the list above because it is subject to reservation conflicts.*

For SBC devices, the following additional commands are not subject to access controls:

1. READ CAPACITY;
2. START/STOP UNIT, START equal one, POWER CONDITION equal zero;
3. SET LIMITS(10), SET LIMITS(12).

For SMC devices, the following additional commands are not subject to access controls:

1. READ ELEMENT STATUS, CURDATA equal one;
2. READ ELEMENT STATUS ATTACHED, CURDATA equal one.

2.1 Identifying initiators

Initiators are identified to the device server by either a TransportID (as defined in the transport-layer protocol standard) or by an AccessID. An AccessID is registered with the device server by an ACCESS ID REGISTER service action. The AccessID is intended to be an identifier of the initiator's host machine. All ports on the host machine can (and should) share this same identifier.

If the device server receives an AccessID from an initiator port, it shall perform the following functions:

1. If the AccessID has been granted some access rights to the device server, the device server shall maintain in volatile memory the association of this AccessID with the port. Subsequent commands from that port can then be handled subject to the access rights of the associated AccessID.
2. If the AccessID has no specific access rights to the device, the device server shall maintain a volatile record that an AccessID was registered from that port. Optionally, the device server can maintain the AccessID/port association in this case.

   NOTE1: When a configuration command to a device server grants access rights to an AccessID that previously had none, a host that has registered that AccessID through some port may need to re-register in order to make use of the granted access. This happens if the device server does not retain the port/AccessID associations for AccessIDs that have no access rights. In this case, the sense data from the device server will not indicate to the host the need to re-register. The re-registration can be triggered, if necessary, by setting the parameters in the configuration command to invoke the FLUSH action of port/AccessID association table and registered ports list at the device server.

The port/AccessID association table may be many-to-one (e.g., if the host has multiple initiator ports). The port/AccessID mapping should never be one-to-many as a single initiator should only have a single AccessID. This port/AccessID association should not be stored in non-volatile memory; it should be invalidated after a power-cycle of the device server.

If any event on the network causes the device server to question whether the initiator at a given port has changed, it should remove this port/AccessID association from its table, thereby restricting access to the device server by that initiator port. The initiator should detect this change of state at the next command failure and can then reissue the ACCESS CONTROL OUT command with ACCESS ID REGISTER service action followed by a retry of the failed command.[1]

If an access right is granted on the basis of a TransportID which might be affected by such network changes (e.g., in FCP, an identifier which include the N_Port identifier), then such changes shall always revoke that access right.

---

1. We do not require that initiators maintain this network state information. An initiator might do this so that it can issue the ACCESS ID REGISTER service action as needed without the retry requirement. This option is beyond the scope of this standard.

If an access right is granted on the basis of a TransportID which contains only persistent components which are not affected by network reconfigurations (such as WWPortName or WWNodeName in FCP), the device server shall reestablish the port/TransportID association and thereby restore the access right.

> NOTE2: The detection process required here is similar to that in the following requirement from FCP-2, rv 02, 5.3): “the relationship between address identifier of the initiator and a persistent reservation for a logical unit can be adjusted as defined in SPC-2 during those reconfiguration events that may change the S_ID of the initiator”. (In FCP, the initiator port identifier is the S_ID.) However, in this proposal, the device server does not maintain this association but severs it until a new ACCESS ID REGISTER service action is received. This has two purposes. First, it prevents physical reconfiguration of initiator ports (say, moving an HBA from one initiator machine to another) from exposing protected data. Second, it associates the access protection with initiator machines and not with the port or node (or HBA). In particular, in Fibre Channel, a PRLO or LOGO (either explicit or implicit) should invalidate port/AccessID associations.
>
> On the other hand, for those access rights based on WWNs, it is exactly this detection and “relationship between address identifier of the initiator... reconfiguration events that may change the S_ID of the initiator” which enables the device server to autonomously reestablish the port/WWN/access relationship.

Additionally, there is provision for vendor-specific host identification placeholders (see 3.1.1.1.3).

2.2 Granting and revoking access rights

There are two service actions defined for configuring access rights and controls at a device server.

A service action (MANGE ACL) can grant or revoke access rights for a specific scope to an initiator based on the AccessID identifier or TransportID identifier (if applicable). The same service action can disable or reenable access controls for a specific scope. It can request the access controls persist through power loss, or disable this function.

A service action (PROXY ACCESS) can grant or revoke access rights for a specific scope to a third party initiator or host. This service action is valid only if the requesting initiator already has access rights to the specific scope. That is, an initiator can extend its own existing rights to another initiator. This allows an initiator to autonomously create an access right for a third party to facilitate additional services such as third party copy.

An initiator’s access rights to a specific scope on a device server are the logical ’or’ of all rights granted under both MANAGE ACL and PROXY ACCESS for any identifier which corresponds to that initiator. For example, an initiator (port) may have rights granted under MANAGE ACL action under both its registered AccessID and TransportID. Similarly, it may have multiple proxy rights granted by other initiators under either the same or different identifiers which correspond to the requesting initiator. Revocation of that initiator’s access rights occurs only when all such access rights have been revoked.

When an initiator’s access rights to a logical unit are revoked, the following cleaning actions shall be applied:

1. all tasks from that initiator on that logical unit shall be aborted;
2. any CA or ACA state for which this is the faulting initiator shall be cleared;
3. any pending UNIT ATTENTIONs for that initiator shall be cleared;
4. any reservations on that logical unit created using RESERVE and held by that initiator are cleared.

> NOTE: There is no directive here to clear any Persistent Reservations created and held by that initiator. The PREEMPT and PREEMPT AND ABORT service actions are available for clearing this reservation and this action should be left to the initiators participating in the reservation protocol.

2.3 Preserving access controls

A device server is required to maintain in non-volatile form an access control state flag for each logical unit which indicates whether that logical unit is unconstrained (access controls are disabled) or constrained in any way (e.g., access controls enabled for the entire logical unit or for any element within the logical unit). These state flags shall persist through power cycles.

The application client may request that the device server preserve the entire access control list including the Access Controls Generation value across power cycles by requesting the Persist Through Power Loss (PTPL) capability. This is done by setting the PTPL field in the MANAGE ACL service action parameter data. Support for this feature is optional but recommended.

AUTHOR'S NOTE: *We use PTPL instead of APTPL to avoid confusion with the APTPL of PERSISTENT RESERVATIONS.*

If the device server does not support this feature or if the PTPL feature is disabled, it shall perform the following functions after a power off period where the previous access control information is lost:

1. for each logical unit, if the access control state flag indicates that access controls were enabled prior to power off, all access-restricted commands are blocked from all initiators to all scopes within the logical unit until new access controls are established at the device server;
2. for each logical unit, if the access control state flag indicates that access controls were disabled prior to the power off, no access controls are enforced for the logical unit;
3. reset the Access Controls Generation value to zero.

If the device server's non-volatile memory is not ready (either to read the access control state flags or the access control table under the PTPL state), the device server will return CHECK CONDITION, a sense key of NOT READY and additional sense data as defined in the TEST UNIT READY command (see SPC-2, rev 11, 7.27) for all access-restricted commands.

If the device server supports the PTPL feature, and the application client requests it, only those access rights granted via a MANAGE ACL service action and the Access Controls Generation value shall persist across a power cycle; proxy access rights shall not.

A device server's access control list, the port/AccessID association table and the registered port list shall not be directly affected by task management functions such as TARGET RESET. Protocol- and interconnect-specific reset events may, however, cause these to be cleared. Specifically, an event that will cause Persistent Reservations (with APTPL not set) to be cleared shall cause the port/AccessID and registered port tables to be cleared, and shall also cause the access control list (and Access Controls Generation value) to be cleared when the PTPL state is not set.

2.4 Reporting access controls

There are two ways to request a report from the device server about access controls.

A service action (REPORT ACL) shall report all access control information for the entire device server independent of the requesting initiator.

In this case, the information includes the following:

1. the number of access control entries currently managed at the device server;
2. the list of scopes for which access controls are currently enabled;
3. the list of scope/Initiator Identifier combinations for which access is currently granted; this includes proxy entries.

The requesting application client may compare the results of this data with the results of a REPORT LUNS and (if applicable) READ ELEMENT STATUS command to determine all scopes for which access controls are currently disabled.

A related service action (REPORT INITIATOR ACL) shall return information only relevant to the requesting initiator.

2.5 Verifying access rights for initiators

Access-restricted commands from a given initiator port are validated in the following manner.

If any of the these conditions hold with respect to the scope defined in the command, the command is handled in the usual way:

1. access controls are disabled;
2. the port has an active proxy access right and the command is not an ACCESS CONTROL OUT with PROXY ACCESS service action;
3. the port has a TransportID (if applicable) which has access rights (proxy or non-proxy);
4. the port has had an AccessID registered and the AccessID has access rights (proxy or non-proxy);
5. the port has an access right granted via a vendor-specific initiator IDENTIFIER TYPE (see 3.1.1.1.3).

If none of these conditions hold, the device server shall transfer no data and shall respond with CHECK CONDITION, sense key ILLEGAL REQUEST and additional sense code of ACCESS DENIED. The additional sense code qualifier is set according the following:

1. if the port has a registered AccessID and neither the AccessID, nor any TransportID nor any vendor-specific identifier for that port has access rights, then the ASCQ is set to INITIATOR NOT AUTHORIZED;
2. if the port has not registered an AccessID, then the ASCQ is set to INITIATOR NOT REGISTERED.

This last case should cause the initiator to issue the ACCESS ID REGISTER service action (if the initiator has an AccessID assigned) and then retry the failed command.

2.5.1 Access rights to elements

For a device server with elements, the device server may optionally allow for access controls at the element level within a given logical unit. Note that access controls can be disabled either at the full logical unit and/or at each element within the logical unit.

For a device server which supports element-based access controls, the following rules apply.

1. An access right granted an initiator to an element within a logical unit also grants that initiator access to the logical unit.
2. An initiator request to a logical unit which does not reference any specific elements within the logical unit (e.g., READ BUFFER, INITIALIZE ELEMENTS) is restricted only by access controls at the logical unit.
3. An initiator request to a logical unit which references one or more particular elements is restricted by the access controls at all of the elements referenced.

2.6 Access Control Service Actions

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

TABLE 1. Access Control Commands and Service Actions

Code	Name	Section
ACCESS CONTROL IN (OPCODE XXh)		
00h	REPORT ACL	3.1.1
01h	REPORT INITIATOR ACL	3.1.2
02h-0Fh	Reserved	
10h-1Fh	Vendor-specific	
ACCESS CONTROL OUT (OPCODE XXh)		
00h	ACCESS ID REGISTER	4.1.1
01h	MANAGE ACL	4.1.2
02h	PROXY ACCESS	4.1.3
03h-0Fh	Reserved	
10h-1Fh	Vendor-specific	

2.7 Access Control Additional Sense Codes

Table 2 contains a list of the Additional Sense Code and Additional Sense Code Qualifiers relevant to access controls.

TABLE 2. Access Control Additional Sense Codes and Qualifiers

ASC	ASCQ	Name	Function
XXh	00h	ACCESS DENIED	Initiator is not sufficiently authorized to make the request.
XXh	XXh	ACCESS DENIED - INITIATOR NOT REGISTERED	Initiator has not sent an ACCESS ID REGISTER service action.
XXh	XXh	ACCESS DENIED - INITIATOR NOT AUTHORIZED	A registered initiator has access permissions insufficient for the requested command
XXh	XXh	ACCESS DENIED - INVALID GENERATION KEY	The Generation key value is does not match the Access Controls Generation value maintained at the device server
XXh	XXh	INSUFFICIENT ACCESS CONTROL RESOURCES	The device server has exhausted its resources for access controls

3.0 ACCESS CONTROL IN command

The ACCESS CONTROL IN command (see Table 3) is used to obtain information about initiator access controls that are active within a device server. The command shall be used in conjunction with the ACCESS CONTROL OUT command. This command shall not be affected by reservations, persistent reservations or access controls (with the exception noted below).

This command should only be sent to LUN0 (in the SAM-2 hierarchical addressing scheme). It should be rejected by the device server if addressed to any other LUN with CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID OPCODE.

TABLE 3. ACCESS CONTROL IN command

Byte	Bit 7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
1	RESERVED			SERVICE ACTION				
2	MSB							
9	GENERATION KEY							LSB
10	MSB							
13	ALLOCATION LENGTH							LSB
14	RESERVED							
15	CONTROL							

The actual length of the ACCESS CONTROL IN parameter data is available in a parameter data field. The ALLOCATION LENGTH field in the CDB indicates how much space has been reserved for the returned parameter data.

The GENERATION KEY field is described in the appropriate subclause for each service action.

The ALLOCATION LENGTH shall be at least eight (8), sufficient for the header information. If the Allocation Length is less than eight (8), then device server shall return CHECK CONDITION with sense data ILLEGAL REQUEST and additional sense code of INVALID FIELD IN CDB.

If the ALLOCATION LENGTH is not sufficient to contain the entire parameter data, the first portion of the data shall be returned. This shall not be considered an error. If the remainder of the data is required, the application client should send a new ACCESS CONTROL IN command with a ALLOCATION LENGTH field large enough to contain the entire data.

3.1 ACCESS CONTROL IN Service Actions

The ACCESS CONTROL IN command service actions are defined in Table 4.

TABLE 4. ACCESS CONTROL IN service actions

Code	Name	Description
00h	REPORT ACL	Used by a client application to query the device server's current access control table. See 3.1.1
01h	REPORT INITIATOR ACL	Used by an initiator to get the summary of his access rights at the device server. See 3.1.2
02h-0Fh	Reserved	Reserved
10h-1Fh	Vendor-specific	Vendor-specific

3.1.1REPORT ACL service action (Mandatory)

The REPORT ACL service action of the ACCESS CONTROL IN command is used by an application client to query the complete access control table currently maintained on the device server.

For a REPORT ACL service action, if the GENERATION KEY does not match the Access Controls Generation value maintained at the device server, the device server may return no data and respond with CHECK CONDITION, sense data ILLEGAL REQUEST, additional sense code of ACCESS DENIED and additional sense code qualifier of INVALID GENERATION KEY.

The format of the returned data shall conform to the specification in 3.1.1.1. Active third party Proxy access rights pages shall also be returned by the device server (see 4.1.3).

3.1.1.1REPORT ACL parameter data format

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 ENABLED AND ENTRY PAGE(S) are described in 3.1.1.1.1 and 3.1.1.1.2.

TABLE 5. REPORT ACL parameter data format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	MSB							
3	RESOURCE UTILIZATION							LSB
4	MSB							
7	ADDITIONAL LENGTH (*n*-7)							LSB
8								
n	ENABLED AND ENTRY PAGE(S)							

The RESOURCE UTILIZATION field shall indicate the total number of entries in the device server's access control table currently used (this is the same as the number of Entry pages available to be returned in the parameter data).

The ADDITIONAL LENGTH field contains 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 (see 3.0).

3.1.1.1.1 REPORT ACL parameter data Access Control Enabled page format

The Access Control Enabled page format for the REPORT ACL service action is specified in Table 6.

TABLE 6. REPORT ACL parameter data Access Control Enabled page format

Byte	Bit 7	6	5	4	3	2	1	0
0	PAGE CODE (00h)							
1	PAGE LENGTH (0Eh)							
2	RESERVED							
3	SCOPE				RESERVED			
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB

One such page shall be returned for every scope for which access controls are currently enabled.

The LUN, SCOPE and the SCOPE-SPECIFIC ADDRESS fields specify the LUN and/or element to which this page refers. The valid SCOPE codes are defined in the PERSISTENT RESERVATION IN clause of SPC-2, rev 11, 7.12.3.1. If the SCOPE field indicates the full logical unit (value 0h), then the SCOPE-SPECIFIC ADDRESS field shall be zero. If the SCOPE field indicates element (that is, SCOPE not equal to 0h), then the SCOPE-SPECIFIC ADDRESS field is as specified in the aforementioned clause of PERSISTENT RESERVATION IN.

Multiple access control enabled pages may be returned for each LUN only if element access controls are enabled at that logical unit.

3.1.1.1.2 REPORT ACL parameter data Access Control Entry page format

The Access Control Entry page format for the REPORT ACL service action is specified in Table 7.

TABLE 7. REPORT ACL parameter data Access Control Entry page format

Byte	Bit 7	6	5	4	3	2	1	0
0	PAGE CODE (01h)							
1	PAGE LENGTH (*n*-1)							
2	RESERVED							
3	SCOPE				RESERVED			PROXY
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB
16	RESERVED							
17	RESERVED							
18	IDENTIFIER TYPE							
19	IDENTIFER LENGTH (*n*-19)							
20	MSB							
n	INITIATOR IDENTIFIER							LSB

The LUN, SCOPE and the SCOPE-SPECIFIC ADDRESS fields are described in 3.1.1.1.1 and specify the scope to which the specified initiator has been granted access.

A PROXY value of zero indicates that the access right was created by an ACCESS CONTROL OUT command with MANAGE ACL service action. The PROXY value of one indicates that the access right to the indicated scope was created by an ACCESS CONTROL OUT command with PROXY ACCESS service action by some other initiator.

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields are specified in 3.1.1.1.3. The IDENTIFIER LENGTH field indicates the number of bytes following taken up by the INITIATOR IDENTIFIER.

One Entry page with PROXY bit equal to zero shall be returned for a given scope and INITIATOR IDENTIFIER combination.

One Entry page with PROXY bit equal to one shall be returned for a given scope and INITIATOR IDENTIFIER combination for every third party proxy granted to that identifier.

3.1.1.1.3 Identifier Type and Initiator Identifier

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields in an Access Control Entry parameter page for an ACCESS CONTROL IN command with service action REPORT ACL or an ACCESS CONTROL OUT command with service action MANAGE ACL or PROXY ACCESS have the meaning described in Table 8.

TABLE 8. IDENTIFIER TYPE and INITIATOR IDENTIFIER values.

Code	Description	Length
00h	AccessID	16
01h	TransportID	TRANSPORT-SPECIFIC
02h-7Fh	Reserved	N/A
80h-FFh	Vendor-specific	VS

AUTHOR'S NOTE: *The AccessID Length was chosen to allow an IPv6 style address to be used as an AccessID (this is NOT required). It was suggested that a longer or variable length AccessID be used to allow implementors a richer and more user-friendly name space. The decision here for fixed length of AccessIDs (what goes over the wire) still leaves open the user-interface side behavior. E.g., an implementation could simply create a pairing of user-friendly names and (say) a hash of that to 16 bytes for the over-the-wire AccessID. In other words, it seemed simpler to push this burden onto the application user-interface and not on the target.*

Use of the TransportID is protocol and interconnect-specific. Each SCSI protocol standard shall specify the description of the TransportID structure.

3.1.2REPORT INITIATOR ACL service action (Optional)

The REPORT INITIATOR ACL service action of the ACCESS CONTROL IN command is used by the initiator to request the device server to send a summary of its own access rights. Support for this service action is optional. If the device server does not support this service action, it shall respond with CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID FIELD IN CDB.

The GENERATION KEY field in the CDB for the REPORT INITIATOR service action is reserved.

The format of the returned data should conform to the specification in 3.1.2.1. Also, the device server will not return any initiator-specific entries for an unrestricted scope (that is, where access controls are disabled).

The device server shall respond with the initiator's relevant Access Control entries for any scope for which access controls are enabled. This includes the following:

1. any entries involving the AccessID (IDENTIFIER TYPE=00h) corresponding to the requesting initiator as registered by an ACCESS CONTROL OUT command with service action ACCESS ID REGISTER;

   NOTE: it is the initiator's responsibility to ensure that he has an active AccessID registration prior to issuing this service action in order to get an accurate report.

2. any entries involving TransportIDs corresponding to the requesting initiator (IDENTIFIER TYPE=01h);

3. any entries corresponding to the requesting initiator as identified by any vendor-specific initiator IDENTIFIER TYPE (see Table 8).

AUTHOR'S NOTE: *The returned list does not reflect any scope for which access controls are disabled (and so for which the initiator has access). It's unclear to me how exactly to handle this. It would be easy to include LUNs for which access controls are disabled but to also include element scopes might be excessive as they are disabled by default. An alternative is to include the "Access Control Enabled" pages for all scopes as is done for REPORT ACL. In this way the requesting initiator can tell to which scopes he has open access (that is, the ones NOT listed).*

If the initiator has no access rights at the device server, then the device server shall return only the required header information.

3.1.2.1REPORT INITIATOR ACL parameter data format

The format of the parameter data provided in response to an ACCESS CONTROL IN command with REPORT INITIATOR ACL service action is shown in Table 9. The ENTRY PAGE(S) are described in 3.1.2.1.1.

TABLE 9. REPORT INITIATOR ACL parameter data format

	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	RESERVED							
1	RESERVED							
2	RESERVED							
3	RESERVED							
4	MSB							
7	ADDITIONAL LENGTH (*n*-7)							LSB
8								
n	ENTRY PAGE(S)							

The ADDITIONAL LENGTH field contains 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 (see 3.0).

3.1.2.1.1REPORT INITIATOR ACL parameter data Access Control Initiator Entry page format

The REPORT INITIATOR ACL service action Access Control Initiator Entry parameter data page format is specified in Table 10.

TABLE 10. REPORT INITIATOR ACL parameter data Initiator Entry page format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	PAGE CODE (02h)							
1	PAGE LENGTH (0Eh)							
2	RESERVED							
3	SCOPE				RESERVED			PROXY
4	MSB							
11		LUN						LSB
12	MSB							
15		SCOPE-SPECIFIC ADDRESS						LSB

All fields in this parameter page are as defined in 3.1.1.1.1 and 3.1.1.1.2.

At most one Entry page with PROXY bit equal to zero shall be returned for a given scope, whether or not that initiator has access rights granted under different INITIATOR IDENTIFIERS corresponding to the requesting initiator.

At most one Entry page with PROXY bit equal to one shall be returned for a given scope, whether or not that initiator has access rights granted under mulitple proxies by different initiators or granted under different INITIATOR IDENTIFIERS corresponding to the requesting Initiator.

AUTHOR'S NOTE: *This data is intended to be a brief summary of that initiator's access rights, not a detailed listing.*

4.0 ACCESS CONTROL OUT Command

The ACCESS CONTROL OUT command (see Table 11) is used to request service actions at a device server to limit or grant access to the device server by initiators. The command shall be used in conjunction with the ACCESS CONTROL IN command. This command shall not be affected by reservations or persistent reservations.

This command should only be sent to LUN0 (in the SAM-2 hierarchical addressing scheme). It should be rejected by the device server if addressed to any other LUN with CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID OPCODE.

TABLE 11. ACCESS CONTROL OUT command

Byte	Bit							
	7	6	5	4	3	2	1	0
0	OPERATION CODE (XXh)							
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 data specify the information required to perform a particular access control service action.

4.1 ACCESS CONTROL OUT Service Actions

The ACCESS CONTROL OUT command service actions are defined in Table 12..

TABLE 12. ACCESS CONTROL OUT service actions

Code	Name	Description
00h	ACCESS ID REGISTER	Used by an initiator to register an AccessID at the device server. See 4.1.1
01h	MANAGE ACL	Used by an application client to add, change, remove entries in the device server's access control table. See 4.1.2
02h	PROXY ACCESS	Used by an initiator to grant or revoke a third party access to a scope to which the requesting initiator already has access. See 4.1.3
03h-0Fh	Reserved	Reserved
10h-1Fh	Vendor-specific	Vendor-specific

4.1.1ACCESS ID REGISTER service action (Mandatory)

The ACCESS ID REGISTER service action of the ACCESS CONTROL OUT command is used by an initiator to inform a device server of its AccessID. The device server will use this information to maintain an association between the (transitory) port identifier of the initiator (e.g., FC address or S_ID) and the AccessID. In this way commands coming from a given initiator port can be referred to the correct entry (or

entries) in the access control table. This service action is mandatory if the ACCESS CONTROL OUT command is supported.

The parameter data contains the AccessID. The PARAMETER LIST LENGTH field shall be sixteen (16). If not, then the device server shall return CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID FIELD IN CDB.

The device server will always respond with status GOOD. The device server is required to maintain a temporary (volatile) record that this command was successful from the issuing initiator port. If the AccessID has any access rights at the device server, then the port and AccessID association shall also be maintained (in volatile memory). In this way, the device server can respond with correct sense information (INITIATOR NOT REGISTERED or INITIATOR NOT AUTHORIZED) to subsequent commands.

4.1.2 MANAGE ACL service action (Mandatory)

The MANAGE ACL version of the ACCESS CONTROL OUT command is used by an application client to authorize access or revoke access to a device server by an initiator. This service action adds, changes or removes an entry or multiple entries in the device server's access control table. This service action can also be used to disable or reenable access controls at a specific scope and to otherwise manage the access control state of the device server. This service action is mandatory if the ACCESS CONTROL OUT command is supported.

The PARAMETER LIST LENGTH field indicates the amount of data which the initiator will send to the device server in the Data Out buffer. The structure of the data is as described in 4.1.2.1. If this value is zero, then no data will be transferred. This is not an error condition.

Any of the following conditions in any parameter page or header require the device server to respond with CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID FIELD IN PARAMETER DATA and also make no changes to the device server's access control state:

1. the scope specification is not valid at the device;
2. the SCOPE field indicates an element and the device server does not support element level access controls;
3. the PROXY field is one;
4. the INITIATOR TYPE field indicates an unsupported value;
5. the INITIATOR TYPE=01h and the INITIATOR IDENTIFIER field is inconsistent with the current network configuration;
6. the PTPL bit is one and the device server does not support non-volatile access controls.

If the device server cannot complete the command because it has insufficient resources to implement the command, it shall return a CHECK CONDITION with sense data ILLEGAL REQUEST and additional sense code of INSUFFICIENT ACCESS CONTROL RESOURCES. In this case, the device server shall restore its access control table to the state prior to receiving this command.

If the device server is currently in its default state (namely, has access controls disabled for all scopes and has an Access Controls Generation value of zero), receipt of this command shall first implicitly constrain the device to block all access restricted commands to all logical units, then instantiate the access controls as specified in the parameter data. For devices with elements, access is initially blocked at the full Logical Unit and not at the individual elements.

If a MANAGE ACL service action causes the complete revocation of access rights from an initiator, the actions listed under 2.2 shall be applied to that initiator.

4.1.2.1 MANAGE ACL parameter data format

The format of the parameter data provided for an ACCESS CONTROL OUT command is shown in Table 13. The ENABLE/DISABLE AND ENTRY PAGE(S) are described in 4.1.2.1.1 and 4.1.2.1.2.

TABLE 13. MANAGE ACL parameter data format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	MSB							
7	GENERATION KEY							LSB
8	MSB							
15	NEW GENERATION KEY							LSB
16	RESERVED							
17	RESERVED							PTPL
18	RESERVED				FLUSH	CLEAR	ENABLEDISABLE	
19	RESERVED							
20 *n*	ENABLE/DISABLE AND ENTRY PAGES(S)							

The GENERATION KEY is used to compare with the current Access Controls Generation value maintained at the device server. If the GENERATION KEY in the parameter data does not match the device server's current Access Controls Generation value, the device server may return CHECK CONDITION with sense key ILLEGAL REQUEST, additional sense code ACCESS DENIED and additional sense code qualifier set to INVALID GENERATION KEY and take no other action. If the device server successfully implements the requested service action, the device server resets its Access Controls Generation value to the value specified in the NEW GENERATION KEY field.

The PTPL (Persist Through Power Loss) bit of one instructs the device server to place all non-proxy access control information valid after successfully completing the current service action in non-volatile memory so that it can be restored after power cycles. This includes the Access Controls Generation value. If this feature is not supported by the device server, it shall respond with CHECK CONDITION, sense key ILLEGAL REQUEST and additional sense code INVALID FIELD IN PARAMETER DATA and no changes to the current access controls are instantiated.

If the PTPL bit is zero, the device server shall only maintain in non-volatile memory the access control state flags for each logical unit (see 2.3).

The FLUSH bit of one instructs the device server to flush its current initiator port to AccessID association table and its registered ports list (those ports which successfully completed the ACCESS ID REGISTER service action).

The CLEAR bit of one instructs the device server to completely clear its entire (global) access control table for all access rights (including proxies) and for all scopes defined at the device server. The enable/disable access controls state for all scopes in the device are set according to the value of the ENABLEDISABLE field,

as described in Table 14. The ENABLE/DISABLE AND ENTRY PAGES that follow in the parameter data provide a new access control state. The CLEAR bit of one will also implicitly force a FLUSH action as specified above.

TABLE 14. Global ENABLEDISABLE Codes.

Code	Description
00b	Leave unchanged the existing access control enable/disable for all logical units and all elements (if applicable).
01b	Enable access controls for all logical units and disable access controls for all elements (if applicable).,
10b	Disable access controls for all logical units and all elements (if applicable)
11b	Reserved

NOTE: The device server can easily be restored to its default unconstrained state by a MANAGE ACL service action with parameter data containing only the header with the following fields set: FLUSH=1, CLEAR=1, ENABLEDISABLE=10b, NEW GENERATION KEY=0.

4.1.2.1.1MANAGE ACL parameter data Access Control Enable/Disable page format

The Access Control Enable/Disable page format for the MANAGE ACL service action is specified in Table 15.

TABLE 15. MANAGE ACL parameter data Access Control Enable/Disable page format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	PAGE CODE (00h)							
1	PAGE LENGTH (0Eh)							
2	RESERVED					CLEAR	ENABLEDIABLE	
3	SCOPE				RESERVED			
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB

The LUN, SCOPE and SCOPE-SPECIFIC ADDRESS fields are defined in 3.1.1.1.1.

The CLEAR bit of one instructs the device server to completely clear its access control table for all access rights (including proxies) for the specified scope. After this clear action, the value of the ENABLEDISABLE field dictates the enable/disable controls state for the specified scope as described in Table 16.

TABLE 16. ENABLEDISABLE Codes

Code	Description
00b	Leave unchanged the existing access control enable/disable for the specified scope
01b	Enable access controls for the specified scope
10b	Disable access controls for the specified scope
11b	Reserved

If the parameter data contains two or more Enable/Disable pages with conflicting instructions, the last such page shall take precedence. This is not an error condition.

4.1.2.1.2 MANAGE ACL parameter data Access Control Entry page format

The Access Control Entry page format for the MANAGE ACL service action is given in Table 17.

TABLE 17. MANAGE ACL parameter data Access Control Entry page format

Byte	Bit 7	6	5	4	3	2	1	0
0	PAGE CODE (01h)							
1	PAGE LENGTH (*n*-1)							
2	RESERVED							REVOKE
3	SCOPE				RESERVED			
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB
16	RESERVED							
17	RESERVED							
18	IDENTIFIER TYPE							
19	IDENTIFIER LENGTH (*n*-19)							
20	MSB							
n	INITIATOR IDENTIFIER							LSB

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields are described in 3.1.1.1.3. The IDENTIFIER LENGTH is described in 3.1.1.1.2.

The LUN, SCOPE and SCOPE-SPECIFIC ADDRESS fields are described in 3.1.1.1.1. Access rights to an element within a logical unit grants rights to the logical unit.

The REVOKE bit of zero directs the device server to allow access to the indicated scope by the indicated initiator without access control restrictions (reservation restrictions can still limit access, however).

A REVOKE bit of one directs the device server to remove any matching access control entries (as created by a previously successful MANAGE ACL service action). Revoking access rights to an element within a logical unit revokes access to the logical unit unless other explicit rights are granted. It is not an error condition if there are no matching access control entries.

In the case of an access right granted on the bases of a TransportID which might be invalid after network configurations, the following shall hold. Any change in the network state which causes the device server to logout or otherwise determine that the TransportID may no longer be associated with the original initiator, shall also cause the device server to revoke that access right for that initiator identifier.

If the parameter data contains two or more Entry pages with conflicting instructions, the last such page shall take precedence. This is not an error condition.

4.1.3 PROXY ACCESS service action (Optional)

The PROXY ACCESS service action of the ACCESS CONTROL OUT command is used by an initiator to grant a third party access to a device server to which the requesting initiator already has access or to remove that access.

Support for this service action is optional. If the device server does not support this service action, the server responds with CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID FIELD IN CDB.

The parameter data will contain a list of Access Control Proxy Entry pages as described in 4.1.3.1. There is no header section of the parameter data for this service action.

Any of the following conditions in any parameter page require the device server to respond with CHECK CONDITION, sense key ILLEGAL REQUEST, and additional sense code INVALID FIELD IN PARAMETER DATA and also make no changes to the device server's access control table:

1. the scope specification is not valid at the device;
2. the SCOPE field indicates an element and the device server does not support element level access controls;
3. the PROXY field is zero;
4. the INITIATOR TYPE field indicates an unsupported value;
5. the INITIATOR TYPE=01h and the INITIATOR IDENTIFIER field is inconsistent with the current network configuration.

If the initiator has no access rights to the scope specified in any parameter page, the device server shall return CHECK CONDITION with sense key ILLEGAL REQUEST and additional sense code of ACCESS DENIED and qualifier of INITIATOR NOT AUTHORIZED and also make no changes to the device server's access control table.

Device servers shall treat proxy entries in a manner consistent with the MANAGE ACL service action, with the following exception. Proxy entries will not be maintained in non-volatile memory even if PTPL state of the device servicer is active.

If the device server has no more resource available to instantiate the proxies, it shall return CHECK CONDITION, sense key ILLEGAL REQUEST with additional sense code of INSUFFICIENT ACCESS CONTROL RESOURCES and the access control table is restored to the state prior to receiving this command and service action.

If the PROXY ACCESS service action causes the complete revocation of access rights from an initiator, the actions listed under 2.2 shall be applied to that initiator.

AUTHOR'S NOTE: *Earlier drafts of this proposal included a Number of Requests field in the proxy parameter page, in order to facilitate self-revocation or auto-expiration of proxies. Though some notion of self-revocation has practical value, it is not clear how to completely specify the device server behavior in the presence of multiple proxies. Consequently, this was removed from this draft. Suggestions on how to reinstated self-revocation are welcome.*

4.1.3.1 PROXY ACCESS parameter data Access Control Proxy Entry page format

The Access Control Proxy Entry page format for the PROXY ACCESS service action is given in Table 18.

TABLE 18. PROXY ACCESS parameter data Access Control Proxy Entry page format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	PAGE CODE (02h)							
1	PAGE LENGTH (*n*-1)							
2	RESERVED							REVOKE
3	SCOPE				RESERVED			PROXY
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB
16	RESERVED							
17	RESERVED							
18	IDENTIFIER TYPE							
19	IDENTIFIER LENGTH (*n*-19)							
20	MSB							
n	INITIATOR IDENTIFIER							LSB

The PROXY bit shall be one in this page for the PROXY ACCESS service action.

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields are described in 3.1.1.1.3. The IDENTIFIER LENGTH is described in 3.1.1.1.2.

The LUN, SCOPE and SCOPE-SPECIFIC ADDRESS fields are described in 3.1.1.1.1.

Revocation of a proxy occurs when the REVOKE bit is one. It is not an error condition to receive the revocation if there is no existing proxy for the indicated third party by the requesting initiator.

In the case of a proxy access right granted on the bases of a TransportID which might be invalid after network configurations, the following shall hold. Any change in the network state which causes the device server to logout or otherwise determine that the TransportID may no longer be associated with the original third party initiator, shall also cause the device server to revoke all existing proxies for that third party initiator.

Any access-restricted command by the third party after revocation of his proxy will be treated as if no proxy had ever been in place.

If the parameter data contains two or more Proxy Entry pages with conflicting instructions, the last such page shall take precedence. This is not an error condition.

A. Changes required in SAM-x.

This section contains some changes required in SAM-x to deal with Task Management in the presense of access controls. (Section numbers correspond to SAM-2, rev. 10).

A.1. Changes for the end of section 6.0.

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

1. a task management request of ABORT TASK, ABORT TASK SET or CLEAR ACA shall be unaffected by the presense of access restrictions;
2. a task management request of CLEAR TASK SET or LOGICAL UNIT RESET received from an initiator that is denied access to the logical unit shall cause no change to the logical unit, but will receive a response of FUNCTION COMPLETE.
3. a TARGET RESET task management request shall initiate a logical unit reset as described in 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.

A.2 Additions for section 5.6.6

While the target 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 controls.

B. Changes required in FCP-x.

This section contains the changes required in FCP-x. This includes the description of the TransportID. (Section numbers correspond to FCP-2, rev. 02).

B.1. Specification of the TransportID

The TransportID structure is 24 bytes long and is described in Table 19.

TABLE 19. TransportID for FCP.

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

A PT_VAL bit of one indicates that the N_PORTID field is valid. Similarly, the PA_VAL, PN_VAL and NN_VAL bits of one indicate that the corresponding PROCESS ASSOCIATOR, WWPORTNAME and WWNODENAME fields, respectively, are valid. A value of zero for any of these bits indicate that the corresponding field is invalid and should be ignored. In the case of a private loop, the N_PORTID should consist only of the AL_PA in the LSB byte of the field (the other bytes should be zero).

For the ACCESS CONTROL OUT command, the following apply. In the MANAGE ACL service action, the PT_VAL and PA_VAL bits must be zero. In the PROXY ACCESS service action, there are no such restrictions.

B.2. Changes to 6.3

CHANGE:

All tasks, reservations, mode page parameters ...that are logged out are not affected.

TO:

All tasks, reservations, mode page parameters, AccessID associations and registrations, and status for image pairs removed by the PRLO operation are set to the state they would have after a SCSI hard reset or power on reset. Only the specified image pairs are logged out. Open exchanges for logged out image pairs shall be terminated by a recovery abort operation. (See 8.1.2.2.) Tasks, reservations, mode page parameters, AccessID associations and registrations, and status for image pairs other than those that are logged out are not affected.

B.3. Additional rows required inTable 4:

TABLE 20.

	POWER	RESETLIP	LOGO, PLOGI	ABTS	PRLI, PRLO	TPRLO	TGTRESET	CLEAR	ABORT	LURESET
Access control list	Y[a]	N	N	N	N	N	N	N	N	N
Port/AccessID association/registration										
For all SCSI initiators	Y	Y	N	N	N	Y	N	N	N	N
Only for SCSI initiator port initiating action	-	-	Y[6]	N	Y	-	N	N	N	N

a. When the most recent PTPL value received by the device server is zero.