Date:Sept 9, 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. We call such partitions SAN Boxes. 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.

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 collaboration between the initiators, it is desirable in this environment, to partition, fence or otherwise restrict access to some storage devices by different hosts. We call such partitions SAN Boxes. The current SAN protocols are not well-suited to this purpose.

In particular, SCSI reservations (either persistent or not) can be usurped or preempted by any other initiator. That is, in the model for reservations, all initiators are peers. Consequently, unless there is a single client application coordinating reservations for *all* initiators in the SAN simultaneously, data integrity and privacy can not be guaranteed. Implementation and management of such an application in a complex, multi-OS, multi-initiator environment is expensive.

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

The essentially new ingredient in our proposal is password authenticated CDBs (without encryption). Device servers maintain a current password (as registered by a client application) and service actions which need authentication shall validate the password in the CDB for successful completion.[1]

Passwords are private at the device server (e.g., can not be queried). This is in contrast to persistent reservation keys which can be discovered by any initiator. Furthermore, these authenticated CDBs need not come from any particular initiator. That is, there is no association at the device server between the password and the initiator which instantiated the password. The password is implicitly associated to a client application which may have access to the device server via any initiator. This again is in contrast to reservations where the key and the initiator are correlated by the device server. Note that there is no requirement for out-of-band communication between this managing client application and the device servers.

A general scenario is the following. A client application (what we call the Partition Access Manager or PAM) has knowledge of all the initiators and target devices on the SAN. It initially configures each target device with a password which will then be used to authenticate all of its future configuration commands. PAM can then instruct a given target device to restrict access to itself by all hosts except those from some small set. Such a set might be a single host. Within the set, data integrity, locking, etc., is managed by existing protocols (like reservations) by client applications managed within the scope of this group. One might say that such a set is a “reservation group”. Hosts outside this group are denied (most) access to the device. In particular, these hosts can not preempt a reservation, issue target resets and the like.

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. Targets can be configured by PAM through any initiator, because the configuration commands are initiator independent and self-authenticating. Furthermore, access restrictions are enforced at the target devices. There are two ways to view this added value. First, new initiators added to the SAN have no access to restricted targets unless expressly added by PAM. Second, initiators need have no special client application 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

1.Password based CDBs might also be used for additional service actions where limited accessibility is required. For example, target configuration changes by a trusted client application or user could easily be folded into these CDBs. This specification allows for vendor-specific service actions for just this purpose.

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

Initiators can be identified either by transport specific identifiers (port numbers, WWNs, etc.), or by a new AccessID as defined in this proposal. For most purposes, the new AccessID is preferred. 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, access rights are naturally associated with the host machine, not the individual initiators (ports/HBAs) on that machine. Port identifiers (e.g., FC N_Port) are transitory; use of these would a require highly available and intrusive presence by PAM within the SAN. The persistent port world wide identifiers are bound typically to the given HBA within a machine and these HBAs are portable. That is they can be moved from machine to machine. Consequently, careful management of the host machine hardware configuration as related to device server access rights based on HBA identifiers requires careful coordination.

The reasons for including the transport specific identifiers will be mentioned in the details that follow.

The intent of the AccessID is to assign a permanent identifier to a given host machine 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.

What follows is a detailed description of the new commands and device server requirements.

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 will reject certain commands from initiators outside the specified set. Initiators can be identified uniquely by either protocol specific mechanisms (e.g., as are used by RESERVE or EXTENDED COPY) or by an access identifier (AccessID) as defined in this proposal.

Application clients may add or remove initiators from the selected set using access control commands. Such commands are (in most cases) authenticated by a password known only to the application client and to the device server. In this way, only a trusted application client can make such changes to the selected set of initiators.

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

1. unrestricted access - all commands are handled in their normal fashion, and
2. 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 (because they are implicitly associated to a client application and not the requesting initiator).

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.

These correspond to the same choices as are available for reservations.

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

1. ACCESS CONTROL OUT - used to create, change or revoke access controls; and
2. ACCESS CONTROL IN - used to query the 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 SIGNED AUTHORIZATIONS service action. This same service action can selectively disable or reenable access controls for any specific scope (logical unit or element within a logical unit).

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.

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 queries information about the device and not the user data. Further, it does not change the state of the device in any way. 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. SET LIMITS(10), SET LIMITS(12).

AUTHOR'S NOTE: *I would leave SET LIMITS off this list since any subsequent commands within the link set will actually fail because of access controls. Again, it is included here only to be consistent with Reservations.*

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.

AUTHOR'S NOTE: *It is not clear to me whether these commands should be free of access restrictions or whether these commands should only return information about elements to which the requesting initiator has access. This point is open to debate.*

2.1 Granting and revoking access rights

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

A service action with valid current password (SIGNED AUTHORIZATIONS) can grant or revoke access rights for a specific scope to an initiator based on either the AccessID identifier or WWN 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 without password (PROXY AUTHORIZATIONS) can grant or revoke access rights for a specific scope to a third party initiator identified by any mechanism other than AccessID (these are private and should not be shared between hosts). These service actions are 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 (i.e., without intervention by PAM) create an access right for a third party to facilitate additional services such as third party copy.

AUTHOR'S NOTE: *In PROXY AUTHORIZATIONS we see one place were transport specific initiator identifiers become useful. The use of transport specific identifiers here parallels that used in both third party reservations and in EXTENDED COPY.*

2.2 Passwords

A device server is required to maintain two passwords (each eight (8) bytes long) and a password state flag in non-volatile memory.

The device server shall maintain a current registered password in non-volatile memory. This password is used to authenticate certain access control commands and service actions at the device server. If such a command is received and the device server's non-volatile memory is not ready, the device server shall return CHECK CONDITION, NOT READY and additional sense data as defined in the TEST UNIT READY command (see SPC-2, rev 11, 7.27).

The device server shall maintain a password state flag to indicate whether the current registered password stored on the device is valid. A password state is valid after successful completion of any SIGNED PASS-

WORD REGISTER service action of type other than UNSET. The invalid state is the default state of the device and after a successful SIGNED PASSWORD REGISTER service action of type UNSET.

The device server shall have a Hardware Password accessible in non-volatile memory. This password is unchangeable and should be operator-readable only by physical access to the device. If the managing client application (PAM) looses knowledge of the current registered password, it can use this hardware password to regain management control of the device server (with appropriate operator assist).

2.3 Preserving access controls

The application client PAM may request that the device server preserve the access controls across power cycles by requesting the Activate Persist Through Power Loss (APTPL) capability. This is done by setting the APTPL field in the SIGNED AUTHORIZATIONS service action parameter data.

Support for this is optional but recommended.

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

1. if the current registered password is valid, all access-restricted commands are blocked from all initiators to all scopes until new access controls are established at the device server;
2. if the current registered password is invalid, access controls are not enforced.

If the device server supports the APTPL feature and its non-volatile memory is not ready, the device server will return CHECK CONDITION, 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.

Only non-proxy access controls should be maintained in non-volatile memory.

2.4 Reporting access controls

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

A service action with valid current password (SIGNED REPORT AUTHORIZATIONS) shall report all access control information for the entire device server independent of the requesting initiator (that is, such a request is coming from PAM).

In this case, the information includes the following:

1. the generation value of the access controls;
2. the number of access control entries currently managed at the device server;
3. the list of scopes for which access controls are currently enabled;
4. the list of scope/initiator combinations for which access is currently granted to the initiator; 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 request without password (REPORT AUTHORIZATIONS) shall return information only relevant to the requesting initiator.

2.5 Identifying initiators

Initiators are identified to the device server in two possible ways. First, they can be identified by transport specific identifiers. These are either volatile (e.g., port identifiers) or persistent (e.g., world wide identifiers). This is the same as is used for reservations or EXTENDED COPY as defined in SPC-2.

The second identification method is by an AccessID 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 (i.e., initiators) on the host machine can (and should) share this same identifier.

Access controls may be granted on the basis of either identification scheme.

If the device server receives an AccessID from an initiator port (via the ACCESS CONTROL OUT command with ACCESS ID REGISTER service action), 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.

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 reset 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 removing 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 transport specific port identifier, then network configuration changes as above shall always revoke that access right.

If an access right is granted on the basis of a transport specific persistent world wide identifier (e.g., in FCP, the WWNodeName), the device server shall reestablish the port/WWN association and thereby restore the access right.

> NOTE1: 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).
>
> 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

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.

S_ID of the initiator" which enables the device server to autonomously reestablish the port/WWN/access relationship.

2.6 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;
3. the port's WWN (if applicable) has any access rights (proxy or non-proxy);
4. the port has had an AccessID registered and the AccessID has access rights.

If none of these conditions hold, the device server responds with CHECK CONDITION, ILLEGAL REQUEST and additional sense code of INITIATOR NOT AUTHORIZED. The additional sense code qualifier is set according the following:

1. if the port has a registered AccessID and the AccessID has no access rights, then the ASCQ is set to INITIATOR ACCESS DENIED;
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.

2.6.1 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.

2.7 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	SIGNED REPORT AUTHORIZATIONS	3.2.1
01h	REPORT AUTHORIZATIONS	3.2.2
ACCESS CONTROL OUT (OPCODE XXh)		
00h	SIGNED PASSWORD REGISTER	4.1.1
01h	ACCESS ID REGISTER	4.1.2
02h	SIGNED AUTHORIZATIONS	4.1.3
03h	PROXY AUTHORIZATIONS	4.1.4

2.8 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	INITIATOR NOT AUTHORIZED	Initiator is not sufficiently authorized to make the request
XXh	XXh	INITIATOR NOT AUTHORIZED - PASSWORD INVALID	Password invalid
XXh	XXh	INITIATOR NOT AUTHORIZED - INITIATOR NOT REGISTERED	Initiator has not sent an ACCESS ID REGISTER command
XXh	XXh	INITIATOR NOT AUTHORIZED - INITIATOR ACCESS DENIED	A registered initiator has access rights insufficient for the requested command
XXh	00h	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 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, ILLEGAL REQUEST, INVALID OPCODE.

TABLE 3. ACCESS CONTROL IN command

	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	OPERATION CODE (XXh)							
1	RESERVED			SERVICE ACTION				
2	MSB							
9	PASSWORD							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 ALLOCATION LENGTH shall be at least twelve (12), sufficient for the header information. If the Allocation Length is less than twelve (12), 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 parameter data general format

The format of the parameter data provided in response to an ACCESS CONTROL IN command is shown in Table 4. The ENABLE/DISABLE AND ENTRY PAGE(S) are described in the appropriate clause for each service action.

TABLE 4. ACCESS CONTROL IN parameter data format

Byte	Bit 7	6	5	4	3	2	1	0
0	MSB							
3		GENERATION						LSB
4	RESERVED							
5	RESERVED							
6	MSB							
7		RESOURCE UTILIZATION						LSB
8	MSB							
11		ADDITIONAL LENGTH (*n*-11)						LSB
12								
n	ENABLED AND ENTRY PAGE(S)							

The GENERATION field shall contain a 32-bit counter maintained by the device server that shall be incremented every time an ACCESS CONTROL OUT command with SIGNED AUTHORIZATIONS service action completes successfully. The counter shall not be incremented by an ACCESS CONTROL IN command, by an ACCESS CONTROL OUT command for any other service action or by an ACCESS CONTROL OUT command that is not performed due to an error condition. The generation value shall be set to zero as part of the power on reset process only if the APTPL state (see 4.1.3.1) does not require non-volatile persistence of the access controls.

The RESOURCE UTILIZATION field is described in the appropriate subclause for each service action.

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).

3.2 ACCESS CONTROL IN Service Actions

The ACCESS CONTROL IN command service actions are defined in Table 5.

TABLE 5. ACCESS CONTROL IN service actions

Code	Name	Description
00h	SIGNED REPORT AUTHORIZATIONS	Used by a client application to query the device server's current access control table. See 3.2.1
01h	REPORT AUTHORIZATIONS	Used by an initiator to get the summary of his access rights at the device server. See 3.2.2
02h-0Fh	Reserved	Reserved
10h-1Fh	Vendor-specific	Vendor-specific

AUTHOR'S NOTE: *The VS values here and in the "out" version are to allow vendors to piggy-back other functions which ought to be "password" protected into the same CDB structure. For example, a vendor might want to put other types of VS-specific configuration operations under this umbrella in order to restrict access to these functions.*

3.2.1 SIGNED REPORT AUTHORIZATIONS service action (Mandatory)

The SIGNED REPORT AUTHORIZATIONS service action of the ACCESS CONTROL IN command is used by an application client to query the access controls currently maintained on the device server.

The PASSWORD field is used by the device server to compare with the server's current registered password. If there is no current registered password or these do not match, the device server will return a CHECK CONDITION with sense data ILLEGAL REQUEST, additional sense code of INITIATOR NOT AUTHORIZED and additional sense code qualifier of PASSWORD INVALID.

The format of the returned data shall conform to the specification in 3.1, 3.2.1.1 and 3.2.1.2. Active third party Proxy Authorizations pages shall also be returned by the device server. See 4.1.4.

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

3.2.1.1 SIGNED REPORT AUTHORIZATIONS parameter data Access Control Enabled page format

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

TABLE 6. SIGNED REPORT AUTHORIZATIONS 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.

3.2.1.2 SIGNED REPORT AUTHORIZATIONS parameter data Access Control Entry page format

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

TABLE 7. SIGNED REPORT AUTHORIZATIONS parameter data Access Control Entry page format

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

The LUN, SCOPE and the SCOPE-SPECIFIC ADDRESS fields are described in 3.2.1.1 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 SIGNED AUTHORIZATIONS 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 AUTHORIZATIONS service action by some other initiator.

The NUMBER OF REQUESTS field is valid only if the PROXY bit is one. In this case, it shall reflect the number of command requests still available to the specified initiator. That is, the number of requests as provided in the PROXY AUTHORIZATIONS service action which created the proxy less the number of access-restricted commands attempted by that initiator. If the number of requests field in the PROXY AUTHORIZATIONS service action was zero, this field shall also be zero. If the PROXY bit is zero, this field shall be zero.

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields are specified in 3.2.1.3.

3.2.1.3 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 SIGNED REPORT AUTHORIZATIONS or an ACCESS CONTROL OUT command with service action SIGNED AUTHORIZATIONS or PROXY AUTHO-

RIZATIONS have the meaning described in Table 8. Any byte positions not specified are reserved and shall be zero. Any Code values not listed are reserved. The INITIATOR IDENTIFIER field is twelve (12) bytes.

TABLE 8. IDENTIFIER TYPE and INITIATOR IDENTIFIER values.

Code	Proxy	Name	Bytes	Convention
00h	0	AccessID	0-11	new
01h	0	Transport WWN	4-11	new
00h	1	(short) third party ID	3	RESERVE/RELEASE
01h	1	(long) third party ID	4-11	RESERVE/RELEASE
02h	1	FC N_Port S_ID	1-3	EXTENDED COPY
03h	1	FC WWN	4-11	EXTENDED COPY
04h	1	FC N_Port S_ID and WWN	1-11	EXTENDED COPY

3.2.1.3.1 Non-proxy initiator identifiers

When the PROXY bit is zero in a parameter page, these fields have the following meanings.

If the IDENTIFIER TYPE field is 00h, then the INITIATOR IDENTIFIER field corresponds to the AccessID as might be registered by an initiator with the ACCESS CONTROL OUT command with ACCESS ID REGISTER service action.

If the IDENTIFIER TYPE is 01h, then the INITIATOR IDENTIFIER field corresponds to a transport layer world-wide unique identifier for this initiator (for FCP, this is the 8 byte Nodename of the initiator). In this case, the first four bytes of the Initiator Identifier shall be zero and the last eight bytes are the WWN.

AUTHOR'S NOTE: *It is a debatable point whether the Transport WWN identifier type is necessary in this proposal. It is the author's feeling that transport specific identifiers are essential for proxy authorizations so including them for non-proxy authoriizations requires no extra burden on the part of the target devices. Furthermore, many existing systems are using transport WWNs for similar access control functions so their use is already accepted.*

3.2.1.3.2 Proxy initiator identifiers

When the PROXY bit is one in the parameter page, the format for the INITIATOR IDENTIFIER corresponds to a third party identifier which is transport-layer specific. The naming convention in this case conforms to a mixture of the convention in the proposed EXTENDED COPY command and the convention of standard RESERVE/RELEASE commands.

AUTHOR'S NOTE: *This mixture of third party identification schemes may be more than is required. It is the author's feeling that the minima/ requirement is for FC_ N_Port S_ID (code 02h) and FC WWN (code 03h) with the extensibility to include other transport protocol identifiers other than FCP. The mixed N_Port and WWN (code 04h) is there only to conform with EXTENDED COPY. The third party IDs (short and long) are there to conform to other third party naming schemes currently in the standards.*

AUTHOR'S NOTE: *The Parallel Interface T_L target descriptor format is not included here because that is a target descriptor and these identifiers are for initiators only.*

Table 9 shows the structure of the INITIATOR IDENTIFIER field in this case.

TABLE 9. Third Party Identifier for Proxies

Byte	Bit							
	7	6	5	4	3	2	1	0
0	RESERVED							
1	MSB							
3		SHORT IDENTIFIER						LSB
4	MSB							
11		LONG IDENTIFIER						LSB

If the IDENTIFIER TYPE field is 00h, then the LONG IDENTIFIER field and the first two bytes of the SHORT IDENTIFIER field are Reserved and the last byte of the SHORT IDENTIFIER corresponds to the one byte Third Party device ID of RESERVE/RELEASE.

If the IDENTIFIER TYPE field is 01h, then the SHORT IDENTIFIER field is reserved and the LONG IDENTIFIER corresponds to the long (8 byte) form of the Third Party Identifier structure in RESERVE/RELEASE.

If the IDENTIFIER TYPE field is 02h, then the SHORT IDENTIFIER field corresponds (at least in FC) to the N_Port S_ID of the third party and the LONG IDENTIFIER field is Reserved.

If the IDENTIFIER TYPE field is 03h, then the SHORT IDENTIFIER field is Reserved and the LONG IDENTIFIER field corresponds to the FC 8 byte WWN (node name) of the third party.

If the IDENTIFIER TYPE field is 04h, then the SHORT IDENTIFIER field is the N_Port S_ID and the LONG IDENTIFIER field corresponds to the FC 8 byte WWN (Nodename) of the third party.

3.2.2 REPORT AUTHORIZATIONS service action (Optional)

The REPORT AUTHORIZATIONS 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, ILLEGAL REQUEST, INVALID FIELD IN CDB.

For this service action, the PASSWORD field in the ACCESS CONTROL IN command CDB is reserved.

The format of the returned data should conform to the specification in 3.1 and 3.2.2.1. No Access Control Enabled pages are to be returned in response to this command. 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 transport-specific WWN associated with the requesting initiator (IDENTIFIER TYPE=01h, non-proxy);
2. any entries involving the AccessID (IDENTIFIER TYPE=00h, non-proxy) associated 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.
3. third party proxy authorizations granted to the requesting initiator (by some other party via the ACCESS CONTROL OUT command with PROXY AUTHORIZATIONS service action) as identified by any of the proxy-specific IDENTIFIER TYPEs in 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 SIGNED REPORT AUTHORIZATIONS. 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.

The RESOURCE UTILIZATION field is reserved for this service action.

3.2.2.1 REPORT AUTHORIZATIONS parameter data Access Control Initiator Entry page format

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

TABLE 10. REPORT AUTHORIZATIONS parameter data Initiator Entry page format

	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	PAGE CODE (02h)							
1	PAGE LENGTH (12h)							
2	RESERVED							
3	SCOPE				RESERVED			PROXY
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB
16	MSB							
17	NUMBER OF REQUESTS							LSB
18	RESERVED							
19	RESERVED							

All fields in this parameter page are as defined in 3.2.1.1 and 3.2.1.2.

AUTHOR'S NOTE: *The only difference between this page and the "signed" version is the omission of the* IDENTIFIER TYPE *and* INITIATOR IDENTIFIER *fields. These could be put back in for the sake of uniformity but they might as well be reserved since the initiator already knows these values. It seems simpler just to leave them out.*

4.0 ACCESS CONTROL OUT Command

The ACCESS CONTROL OUT command (see Table 10) 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, ILLEGAL REQUEST, 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	MSB							
9	PASSWORD							LSB
10	MSB							
13	PARAMETER LIST LENGTH							LSB
14	RESERVED					PASSWORD TYPE		
15	CONTROL							

If the ACCESS CONTROL OUT command is attempted, but there are insufficient device server resources to complete the operation, the device server shall return CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to INSUFFICIENT ACCESS CONTROL RESOURCES. No changes to the device server's access controls shall be made.

The ACCESS CONTROL OUT command contains fields that specify an access control service action. The PASSWORD and PASSWORD TYPE fields are defined in 4.1.1.

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

When processing the ACCESS CONTROL OUT command service actions, the device server shall increment the generation value as specified in 3.1.

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	SIGNED PASSWORD REGISTER	Used by a client application to register a new, change or remove a password with the device server. See 4.1.1
01h	ACCESS ID REGISTER	Used by an initiator to register an AccessID at the device server. See 4.1.2
02h	SIGNED AUTHORIZATIONS	Used by an application client to add, change, remove entries in the device server's access control table. See 4.1.3
03h	PROXY AUTHORIZATIONS	Used by an initiator to temporarily grant or revoke a third party access to a scope to which the requesting initiator already has access. See 4.1.4
04h-0Fh	Reserved	Reserved
10h-1Fh	Vendor-specific	Vendor-specific

4.1.1 SIGNED PASSWORD REGISTER service action (Mandatory)

The SIGNED PASSWORD REGISTER service action of the ACCESS CONTROL OUT command is used by an application client to register a password with the device server. This password is used with some service actions of the ACCESS CONTROL IN and ACCESS CONTROL OUT commands to authenticate the managing client application. Passwords are eight (8) bytes long. This service action is mandatory if the ACCESS CONTROL OUT command is supported.

The PASSWORD TYPE field in the command CDB determines how the PASSWORD field in the CDB is used. Table 13 describes the different values and their meaning.

TABLE 13. PASSWORD TYPE Codes

Code	Name	Description
000b	CURRENT	Compare the PASSWORD field in the command to the current password registered at the device server by the last successful SIGNED PASSWORD REGISTER service action.
001b	INITIAL	Valid only if no current password is registered at the device server. No password comparison takes place.
010b	HARDWARE	Compare the PASSWORD field in the command to the hardware password stored permanently at the device server.
011b	UNSET	Compare the PASSWORD field in the command to the current registered password (see CURRENT)
100b-111b	Reserved	Reserved

Any of the following conditions cause the device server to return CHECK CONDITION with sense data set to ILLEGAL REQUEST and additional sense code of INVALID FIELD IN CDB.

1. The PASSWORD TYPE field is set to 001b (INITIAL) and a current password has already been registered at the device server.
2. The PASSWORD TYPE field is set to any Reserved value.
3. The PARAMETER LIST LENGTH field is not zero when the PASSWORD TYPE field is 011b (UNSET) and does not equal eight (8) in all other cases.

Any of the following conditions cause the device server to return CHECK CONDITION with sense data set to ILLEGAL REQUEST, additional sense code of INITIATOR NOT AUTHORIZED and additional sense code qualifier of PASSWORD INVALID.

1. The PASSWORD TYPE field is set to 00b (CURRENT) or 011b (UNSET) and the PASSWORD field does not match the current password registered at the device server.
2. The PASSWORD TYPE field is set to 010b (HARDWARE) and the PASSWORD field does not match the hardware password stored permanently at the device server.

If the PASSWORD TYPE field is set to 011b (UNSET), then the PARAMETER LIST LENGTH field shall be set to zero and no parameter data is required. If the password match check is valid, the device server will remove any existing access controls (including proxies) at the device server for all scopes, remove the existing current registered password and restore its access control state to its default unconstrained state. In particular, the device server shall act as if no successful SIGNED PASSWORD REGISTER service action had occurred. The password state is set to invalid.

If the PASSWORD TYPE field is set to any other acceptable value, the PARAMETER LIST LENGTH field of the command shall equal eight (8) and the parameter data shall contain a NEW PASSWORD as specified in Table 14. If the password match check (as specified in Table 13) is valid, then the current registered password is set to the value in the NEW PASSWORD of the parameter data and the password state is set to valid. Passwords shall be stored in non-volatile storage on the device servers so they survive resets and power cycles.

TABLE 14. SIGNED PASSWORD REGISTER parameter data format

	Bit							
Byte	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	MSB							
7	NEW PASSWORD							LSB

When the device server gets its registered password set with the Initial bit, the device server remains unconstrained and no access checking is performed.

After a Hardware Password reset, the device server shall maintain its current access controls.

4.1.2 ACCESS 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 twelve (12)-byte AccessID. The PARAMETER LIST LENGTH field shall equal twelve (12). If not, then the device server shall return CHECK CONDITION, ILLEGAL REQUEST, 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 ACCESS DENIED) to subsequent commands.

4.1.3 SIGNED AUTHORIZATIONS service action (Mandatory)

The SIGNED AUTHORIZATIONS version of the ACCESS CONTROL OUT command is used by a client application 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 PASSWORD TYPE field in the CDB is reserved for this service action.

The PASSWORD field is used by the device server to compare with its current registered password. If there is no current registered password or if these do not match, the device server will return a CHECK CONDITION with sense code ILLEGAL REQUEST and additional sense code of INITIATOR NOT AUTHORIZED and additional sense code qualifier of PASSWORD INVALID.

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 4.1.3.1. 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, ILLEGAL REQUEST, 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 IDENTIFIER field is inconsistent (see 3.2.1.3).
5. The APTPL 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 no more resources for access control entries, 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 acting as a completely unconstrained device, receipt of this command will first implicitly constrain the device to block all access, 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.

4.1.3.1 SIGNED AUTHORIZATIONS parameter data general format

The format of the parameter data provided for an ACCESS CONTROL OUT command is shown in Table 15. The ENABLE/DISABLE AND ENTRY PAGE(S) are described in 4.1.3.2 and 4.1.3.3.

TABLE 15. SIGNED AUTHORIZATIONS parameter data format

Byte	Bit							
	7	6	5	4	3	2	1	0
0	RESERVED							
1	RESERVED							
2	RESERVED					CLEAR	FLUSH	APTPL
3	RESERVED							
4 *n*	ENABLE/DISABLE AND ENTRY PAGES(S)							

The CLEAR bit of one instructs the device server to completely clear its access control table for all access rights (including proxies) and reenable access controls for all scopes. 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 below.

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

The APTPL (Activate Persist Through Power Loss) bit instructs the device server to place the (non-proxy) access controls in place after successfully completing the current service action in non-volatile memory so that it can be restored after power cycles. If this feature is not supported by the device server, it shall respond with CHECK CONDITION, ILLEGAL REQUEST, INVALID FIELD IN PARAMETER DATA and no changes to the current access controls are instantiated.

4.1.3.2 SIGNED AUTHORIZATIONS parameter data Access Control Enable/Disable page format

The Access Control Enable/Disable page format for the SIGNED AUTHORIZATIONS service action is specified in Table 16.

TABLE 16. SIGNED AUTHORIZATIONS 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 3.2.1.1.

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.

The values and interpretations of the ENABLEDISABLE field are described in Table 17.

TABLE 17. 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.

4.1.3.3 SIGNED AUTHORIZATIONS parameter data Access Control Entry page format

The Access Control Entry page format for the SIGNED AUTHORIZATIONS service action is given in Table 18.

TABLE 18. SIGNED AUTHORIZATIONS parameter data Access Control Entry page format

	Bit							
Byte	7	6	5	4	3	2	1	0
0	PAGE CODE (01h)							
1	PAGE LENGTH (1Eh)							
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	RESERVED							
20	MSB							
31	INITIATOR IDENTIFIER							LSB

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields are described in 3.2.1.3.

The LUN, SCOPE and SCOPE-SPECIFIC ADDRESS fields are described in 4.1.3.2. 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 SIGNED AUTHORIZATIONS service action), thereby denying access to the scope (unless access controls are disabled for that scope). 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.

If the parameter data contains two or more Entry pages with conflicting instructions, the last such page shall take precedence.

4.1.4 PROXY AUTHORIZATIONS service action (Optional)

The PROXY AUTHORIZATIONS service action of the ACCESS CONTROL OUT command is used by an initiator to grant a third party temporary access to a device server 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, ILLEGAL REQUEST, INVALID FIELD IN CDB.

The parameter data will contain a list of Access Control Proxy Entry pages as described in 4.1.3.3. 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, ILLEGAL REQUEST, 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 IDENTIFIER field is inconsistent (see 3.2.1.3).

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

With the exception of the NUMBER OF REQUESTS value (see 4.1.4.1), device servers shall treat proxy entries in a manner consistent with the SIGNED AUTHORIZATIONS service action, with the following exception. Proxy entries will not be maintained in non-volatile memory if APTPL state of the device service is active.

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

4.1.4.1 PROXY AUTHORIZATIONS parameter data Access Control Proxy Entry page format

The Access Control Proxy Entry page format for the PROXY AUTHORIZATIONS service action is given in Table 19.

TABLE 19. PROXY AUTHORIZATIONS 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 (1Eh)							
2	RESERVED							REVOKE
3	SCOPE				RESERVED			PROXY
4	MSB							
11	LUN							LSB
12	MSB							
15	SCOPE-SPECIFIC ADDRESS							LSB
16	MSB							
17	NUMBER OF REQUESTS							LSB
18	IDENTIFIER TYPE							
19	RESERVED							
20	MSB							
31	INITIATOR IDENTIFIER							LSB

The PROXY bit shall be one in this page for the PROXY AUTHORIZATIONS service action. This value dictates the interpretation of the IDENTIFIER TYPE field.

The IDENTIFIER TYPE and INITIATOR IDENTIFIER fields are described in 3.2.1.3.

The LUN, SCOPE and SCOPE-SPECIFIC ADDRESS fields are described in 4.1.3.2.

The NUMBER OF REQUESTS field indicates the maximum number of access-restricted commands that the third party may request at the device server. If this field is not zero, the device server shall maintain a counter of the number of requests received from the third party. When the counter equals the NUMBER OF REQUESTS value, the third party proxy is revoked by the device server. A value of zero for this field indicates unlimited number of requests are accepted and no counter is required.

Revocation of a proxy occurs when the REVOKE bit is one. The NUMBER OF REQUESTS field is ignored in this case. 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.

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.

In the case of third party identifiers based on any addressing method other than FC WWN (that is, all IDENTIFIER TYPE except 03h and 04h), the following shall hold. Any change in the network state which causes the device server to logout or otherwise determine that the third party device ID may no longer be associated with the original third party initiator, the device server will revoke all existing proxies for that third party initiator.

For FC WWN third party identifiers, IDENTIFIER TYPE (03h), the device server shall maintain the proxy through network configuration changes which might change the FC N_Port S_ID for that initiator until the proxy is revoked explicitly or implicitly.

For FC N_Port S_ID and WWN third party identifiers, IDENTIFIER TYPE (04h), the following shall hold. If the device server cannot confirm the association between the N_Port and WWN, the command will terminate with CHECK CONDITION status and sense data ILLEGAL REQUEST, INVALID FIELD IN PARAMETER DATA. If the command is accepted, the device server shall track configuration changes that affect the S_ID of the specified WWN initiator and maintain the proxy authorization until revoked explicitly or implicitly.

If the parameter data contains two or more Proxy Entry pages with conflicting instructions, the last such page shall take precedence.