To: T10 Technical Committee
From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)
Date: 18 October 2001
Subject: SPC-3 Letting persistent reservations ignore initiator ports

Revision History

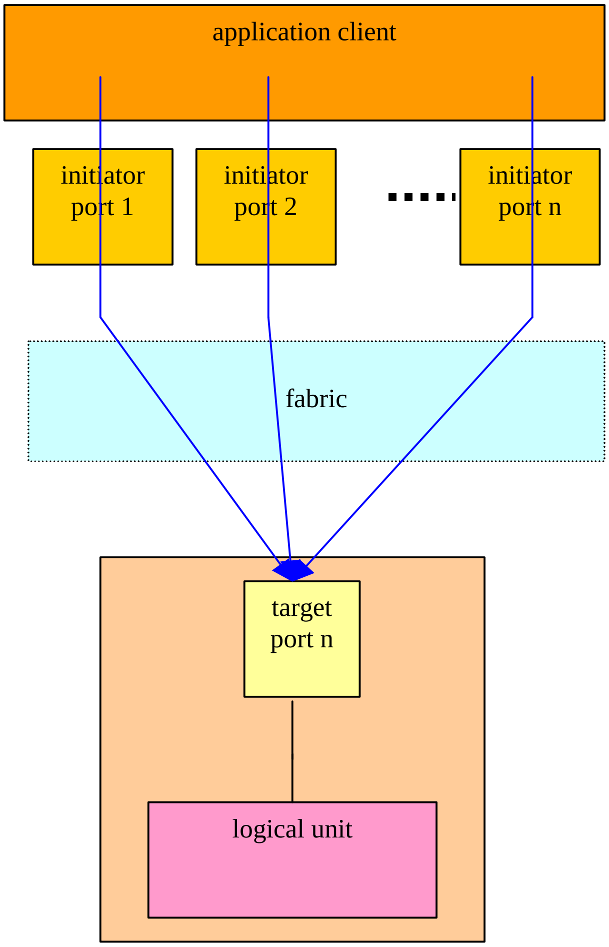
Revision 0 (7 March 2001). First revision. Not complete, but as presented in CAP working group.
Revision 1 (24 April 2001). Reflects input from the March CAP working group. Settles on separate PR OUT service action to register an initiator ID.
Revision 2 (18 October 2001): Incorporated input from May CAP working group. Changed "initiator device name" to "application name." Brought up to date with 01-099r4.

Related Documents

T10/spc3r01 – SCSI Primary Commands 3 revision 1 (by Ralph Weber)
T10/00-232r9 – Asymmetric target behavior (by Ken Moe)
T10/01-099r4 - SPC-3 Letting persistent reservations ignore initiator ports

Overview

SPC-2 requires that logical units remember the initiator port through which a reservation was made. An application client (a thread of an application) wishing to make a reservation needs to run the PERSISTENT RESERVE command through each initiator port that can route to the logical unit.



In many cases, the logical unit doesn't care which initiator port the reservation came through. It just wants to distinguish between applications. Requiring reservation commands for each initiator port burdens the applications with issuing extra commands and burdens the logical unit with extra non-volatile storage. Cases include a single host with multiple ports and multiple hosts in a cluster.

In some clusters of high-end systems, there may be hundreds of initiator ports per node (these machines have lots of PCI busses each holding HBAs), dozens of initiator nodes (8-100 node clusters), all talking to a logical unit through a handful of target ports (4 port targets are common). For example, there may be 256 * 64 * 4 = 65,536 different I_T paths between the same application and logical unit. For an all-registrants type of reservation, this results in 65,536 registration commands, each of which may be slow and require media access.

This proposal allows persistent reservation commands to use the higher level identifier rather than initiator port identifier.

An "application name" is defined that can be passed as data in a new service action for the PERSISTENT RESERVE OUT command. An application would need to run this one time through each initiator port so the target can correlate the provided application name with the dynamic initiator port identifier. Once done, the application only needs to run PR OUT commands through one initiator port, provided that it sets a new USE APPLICATION NAME bit in the PR OUT parameter data to indicate that the application name needs to be used instead of the initiator port identifier.

The application name must be worldwide unique, across all protocols. Clusters which do not have such control should not use the feature.

The application name may equate to the "initiator device name" hinted at by SAM-2.

This does not reduce the amount of fabric traffic; in fact, it goes up a bit (sending n associate commands followed by one registration/reservation vs. sending n registration/reservation commands). However, the associate commands are considered lightweight, and can be implemented much more efficiently by the target. This does reduce target non-volatile storage; it can remember the application name rather than each initiator port being used by that application.

The application name may also be useful for implementing a form of third-party reservations in a persistent reservation environment. The application name for a cluster reservation can be given to a copy manager without requiring it to participate fully in the cluster.

Suggested Changes

Text is from SPC-3 revision 1 as modified by 01-099r4.

5.5.1 Reservations overview

Reservations may be used to allow a device server to execute commands from a selected set of initiators. The device server shall reject commands from initiators outside the selected set of initiators by uniquely identifying initiators using protocol specific mechanisms.

,,,

5.5.3.1 Overview of the Persistent Reservations management method

…

The PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands provide the basic mechanism for dynamic contention resolution in multiple initiator systems using multiple port targets. Before a persistent reservation may be established, an initiator shall register with a device server using a reservation key. Reservation keys are necessary to allow:

a) authentication of subsequent PERSISTENT RESERVE OUT commands;
b) identification of other initiators that are registered;

c) identification of the reservation key(s) that have an associated reservation;
d) preemption of a persistent reservation from a failing or uncooperative initiator; and
e) multiple initiators to participate in a reservation.

The reservation key provides a method for the application client to associate a protocol-independent identifier with an initiator. The reservation key is used in the PERSISTENT RESERVE IN command to identify which initiators are registered and which initiator, if any, holds the reservation. The reservation key is used in the PERSISTENT RESERVE OUT command; to register an initiator, to verify the initiator issuing the PERSISTENT RESERVATION OUT command is registered, and to specify which initiator's registration or persistent reservation to preempt.

Reservation key values may be used by application clients to identify initiators, using application specific methods that are outside the scope of this standard. This standard provides the ability to register no more than one key per initiator/logical unit pair. Multiple initiators may use the same key for a logical unit. An initiator may establish registrations for multiple logical units in a SCSI device using any combination of unique or duplicate keys. These rules provide the ability for an application client to preempt multiple initiators with a single PERSISTENT RESERVE OUT command, but they do not provide the ability for the application client to uniquely identify the initiators using the PERSISTENT RESERVE commands.

5.5.3.x Initiator identifying information [section added by 01-099]

On protocols where initiator port names are required, the initiator identifying information includes the initiator port name and may include the relative port identifier of the target port. All device servers shall default to including the relative port identifier. Device servers which are capable of ignoring the relative port identifier shall support the IGNORE TARGET PORT CAPABLE and IGNORING TARGET PORT bits in the REPORT CAPABILITIES service action of PERSISTENT RESERVE IN and the IGNORE TARGET PORT bit in the SET CAPABILITIES service action of PERSISTENT RESERVE OUT.

On protocols where initiator port names are not required, the initiator identifying information includes both the initiator port identifier and the relative port identifier of the target port.

If the device server supports the use application name mode flag, an application client may identify its application with an application name and use that as the initiator identifying information rather than the initiator port name or initiator port identifier for the port it is using. The application client creates an association between an application name and an initiator port with the PERSISTENT RESERVE OUT command's ASSOCIATE APPLICATION NAME service action. The application client may associate the same application name with several initiator ports. The application client then issues subsequent PERSISTENT RESERVE OUT commands with the USE APPLICATION NAME bit set. The initiator identifying information then includes the application name rather than the initiator port name.

Table xx lists the initiator identifying information for each combination of ignore target port mode and use application name mode.

Table xx. Initiator identifying information.

Protocol support for port names	**Ignore target port mode**	**Use application name mode**	**Initiator identifying information**
protocols where initiator port names are not required	true or false	false	a) initiator port identifier b) relative port identifier of the target port
	true or false	true	a) application name b) relative port identifier of the target port
protocols where initiator port names are required	false	false	a) initiator port name b) relative port identifier of the target port
	false	true	a) application name b) relative port identifier of the target port
	true	false	a) initiator port name
	true	true	a) application name

NOTE nn: The device server may use any internal representation of the relative port identifier of the target port. The identifier is not visible to application clients.

5.5.3.2 Preserving persistent reservations

The application client may request activation of the persist through power loss device server capability to preserve the persistent reservation and registration keys across power cycles by setting the APTPL bit to one in PERSISTENT RESERVE OUT parameter data sent with a REGISTER, or a REGISTER AND IGNORE EXISTING KEY service action.

After the application client enables the persist through power loss capability the device server shall preserve all current and future registrations and persistent reservations associated with the logical unit to which the REGISTER or the REGISTER AND IGNORE EXISTING KEY service action was addressed until an application client disables the persist through power loss capability. The APTPL value from the most recent successfully completed REGISTER or REGISTER AND IGNORE EXISTING KEY service action from any application client shall determine the logical unit's behavior in the event of a power loss.

The device server shall preserve the following information for each registration across any reset, and if the persist through power loss capability is enabled, across any power cycle:
a) initiator identifying information (see 5.5.3.x); and
b) reservation key.

The device server shall preserve the following reservation information across any reset, and if the persist through power loss capability is enabled, across any power cycle:
a) initiator identifying information (see 5.5.3.x);
b) reservation key;
c) scope; and
d) type.
…

5.5.3.4 Registering

…

In response to a PERSISTENT RESERVE OUT with a REGISTER or a REGISTER AND IGNORE EXISTING KEY service action the device server shall perform a registration by doing the following as an uninterrupted series of actions:
a) Process the registration request regardless of any persistent reservations;
b) process the APTPL bit;

c) ignore the contents of the SCOPE and TYPE fields;
d) map the reservation key to the registering initiator using the initiator identifying information (see 5.5.3.x);
e) register the reservation key without changing any a persistent reservation that may exist; and
f) retain the reservation key and associated information.

5.6 Multiple port and multiple initiator behavior

SAM-2 specifies the behavior of logical units being accessed by more than one initiator. Additional target ports provide alternate paths through which the device server may be reached and may also provide connectivity for additional initiators. An alternate path may be used to improve the availability of devices in the presence of certain types of failures and to improve the performance of devices whose other paths may be busy.

If a SCSI target device has more than one target port, the arbitration and connection management among the target ports is vendor specific. If one target port is being used by an initiator, accesses attempted through other target port(s) may:
a) receive a status of BUSY; or
b) be accepted as if the other target port(s) were not in use.
The device server shall indicate the presence of multiple target ports by setting the MULTIP bit to 1 in its standard INQUIRY data.

For the purposes of handling reservations, other initiators are defined as all initiators that do not share the same initiator identifying information.

Only the following operations allow an initiator to interact with the tasks of another initiator, regardless of the service delivery port:
a) the PERSISTENT RESERVE OUT with PREEMPT service action preempts persistent reservations for other initiators (see 5.5.3.6.3);
b) the PERSISTENT RESERVE OUT with PREEMPT AND ABORT service action preempts persistent reservations and all tasks for other initiators (see 5.5.3.6.4);
c) the PERSISTENT RESERVE OUT with CLEAR service action releases persistent reservations and removes reservation keys for all initiators (see 5.5.3.6.5);
d) the TARGET RESET task management function releases reservations established by the reserve/release method and removes all tasks for all initiators for all logical units in the target (see SAM-2). Persistent reservations remain unmodified;
e) the LOGICAL UNIT RESET task management function releases reservations established by the reserve/release method and removes all tasks for all initiators for the addressed logical unit and any logical units issuing from it in a hierarchical addressing structure (see SAM-2). Persistent reservations remain unmodified; and
f) the CLEAR TASK SET task management function removes all tasks for all initiators for the selected logical unit. Most other logical unit states remain unmodified, including MODE SELECT parameters, reservations, and ACA (see SAM-2).

7.10.2 PERSISTENT RESERVE IN service actions

7.10.2.1 Summary of PERSISTENT RESERVE IN service actions

The service action codes for the PERSISTENT RESERVE IN command are defined in table 67.

Table 67 — PERSISTENT RESERVE IN service action codes

Code	Name	Description
00h	READ KEYS	Reads all registered Reservation Keys
01h	READ RESERVATION	Reads the current persistent reservations
02h	REPORT CAPABILITIES	Returns capability information
03h – 1Fh	Reserved	Reserved

7.10.2.x Report Capabilities [section added by 01-099]

The REPORT CAPABILITIES service action requests that the device server return a parameter page indicating capability for various persistent reservation features.

Table 73. REPORT CAPABILITIES parameter data

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	LENGTH (0008h)						
1								(LSB)
2	Reserved				APPLICATION NAME CAPABLE	ELEMENT SCOPE CAPABLE	APTPL CAPABLE	IGNORE TARGET PORT CAPABLE
3	Reserved				USING APPLICATION NAME	Rsvd	PTPL	IGNORING TARGET PORT
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							

The LENGTH field specifies the length in bytes of the parameter data. If the ALLOCATION LENGTH field in the CDB is too small to transfer all of the parameter data, the length shall not be adjusted to reflect the truncation.

An IGNORE TARGET PORT CAPABLE bit of one indicates that the target port is capable of omitting the relative target port identifier in the initiator identifying information. An IGNORE TARGET PORT CAPABLE bit of zero indicates that the target port always includes the relative target port identifier in the initiator identifying information.

An APTPL CAPABLE bit of one indicates that the device server supports the APTPL bit in the PERSISTENT RESERVE OUT command. An APTPL CAPABLE bit of zero indicates that the device server does not support the APTPL bit in the PERSISTENT RESERVE OUT command.

An ELEMENT SCOPE CAPABLE bit of one indicates that the device server supports a SCOPE value of ELEMENT_SCOPE in the persistent reservation commands. An ELEMENT SCOPE CAPABLE bit of zero indicates that the device server does not support a SCOPE value of ELEMENT_SCOPE in the persistent reservation commands.

An IGNORING TARGET PORT bit of one indicates that the target port is not including the relative target port identifier in the initiator identifying information. An IGNORING TARGET PORT bit of zero indicates that the target port is including the relative target port identifier in the initiator identifying information. This capability is controlled with the SET CAPABILITIES service action of PERSISTENT RESERVE OUT (7.11.4).

A PTPL bit of one indicates that the most recent successfully completed REGISTER or REGISTER AND IGNORE EXISTING KEY service action had its APTPL bit set to one (see 7.11.3). A PTPL bit of zero indicates that the most recent successfully completed REGISTER or REGISTER AND IGNORE EXISTING KEY service action had its APTPL bit set to zero (see 7.11.3).

The APPLICATION NAME CAPABLE bit indicates whether the device server supports the USE APPLICATION NAME bit in the CDB and the ASSOCIATE APPLICATION NAME and UNASSOCIATION APPLICATION NAME service actions in the PERSISTENT RESERVE OUT command.

The USING APPLICATION NAME bit indicates whether the use application name mode is enabled (see 5.5.3.x).

7.11.1 PERSISTENT RESERVE OUT command introduction

…

Table 73. PERSISTENT RESERVE OUT command

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (5Fh)							
1	Rsvd	Rsvd	~~Rsvd~~ USE APPLICATION NAME	SERVICE ACTION				
2	SCOPE				TYPE			
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	PARAMETER LIST LENGTH (18h)							
8								
15	CONTROL							

The USE APPLICATION NAME bit indicates whether the device server shall use the application name as part of the initiator identifying information rather than the initiator port name or initiator port identifier (see 5.5.3.x). If the USE APPLICATION NAME bit is set to one while the application name mode flag is false or set to zero while the application name mode flag is true, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN CDB.

…

7.11.2 PERSISTENT RESERVE OUT Service Actions

When processing the PERSISTENT RESERVE OUT service actions, the device server shall increment the generation value as specified in 7.10.3.

The PERSISTENT RESERVE OUT command service actions are defined in table 74.

Table 74 — PERSISTENT RESERVE OUT service action codes

Code	Name	Description	GENERATION field incremented (see 7.10.3)	Parameter list
00h	REGISTER	Register a reservation key with the device server (see 5.5.3.4).	Yes	7.11.3
01h	RESERVE	Creates a persistent reservation having a specified SCOPE and TYPE (see 5.5.3.5). The SCOPE and TYPE of a persistent reservation are defined in 7.10.4.2 and 7.10.4.3.	No	7.11.3
02h	RELEASE	Releases the selected reservation for the requesting initiator (see 5.5.3.6.2).	No	7.11.3
03h	CLEAR	Clears all reservation keys and all persistent reservations (see 5.5.3.6.5).	Yes	7.11.3
04h	PREEMPT	Preempts persistent reservations from another initiator (see 5.5.3.6.3).	Yes	7.11.3
05h	PREEMPT AND ABORT	Preempts persistent reservations from another initiator and aborts all tasks for all initiators registered with the specified reservation key (see 5.5.3.6.3 and 5.5.3.6.4).	Yes	7.11.3
06h	REGISTER AND IGNORE EXISTING KEY	Register a reservation key with the device server (see 5.5.3.4).	Yes	7.11.3
07h	SET CAPABILITIES	Set persistent reservation capabilities	No	7.11.4
08h	ASSOCIATE APPLICATION NAME	Associate an application name with the initiator port in the device server	No	7.11.5
09h	UNASSOCIATE APPLICATION NAME	Remove the association of an application name with the initiator port in the device server	No	7.11.5
~~07h~~ 0Ah – 1Fh	Reserved			

~~The parameter list values for each service action except for SET CAPABILITIES are specified in 7.11.3. The parameter list values for SET CAPABILITIES are described in 7.11.4.~~

7.11.3 PERSISTENT RESERVE OUT parameter list

The parameter list required to perform the PERSISTENT RESERVE OUT command for all service actions except the SET CAPABILITIES, ASSOCIATE APPLICATION NAME, and UNASSOCIATE APPLICATION NAME service actions is defined in table 75. ~~All fields~~The first 24 bytes shall be sent on all such PERSISTENT RESERVE OUT commands, even if some of the fields ~~is~~ are not required for the specified service action and scope values. The APPLICATION NAME field shall only be sent if the use application name flag is true.

Table 75. PERSISTENT RESERVE OUT parameter list

Byte\Bit	7	6	5	4	3	2	1	0
0	RESERVATION KEY							
7								
8	SERVICE ACTION RESERVATION KEY							
15								
16	SCOPE-SPECIFIC ADDRESS							
19								
20	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	IGNORE TARGET PORT	APTPL
21	Reserved							
22	Obsolete							
23								

The obsolete field in Bytes 22 and 23 was defined in a previous standard for use with an obsolete scope (see table 71). If the obsolete scope is not supported Bytes 22 and 23 should be zero.

The RESERVATION KEY field contains an 8-byte value provided by the application client to the device server to identify the initiator that is the source of the PERSISTENT RESERVE OUT command. The device server shall verify that the contents of the RESERVATION KEY field in a PERSISTENT RESERVE OUT command parameter data matches the registered reservation key for the initiator from which the task was received, except for:
a) the REGISTER AND IGNORE EXISTING KEY service action where the RESERVATION KEY field shall be ignored; and
b) the REGISTER service action for an unregistered initiator where the RESERVATION KEY field shall contain zero.

Except as noted above, when a PERSISTENT RESERVE OUT command specifies a RESERVATION KEY field other than the reservation key registered for the initiator the device server shall return a RESERVATION CONFLICT status. Except as noted above, the reservation key of the initiator shall be verified to be correct regardless of the SERVICE ACTION and SCOPE field values.

The SERVICE ACTION RESERVATION KEY field contains information needed for four service actions; the REGISTER, REGISTER AND IGNORE EXISTING KEY, PREEMPT, and PREEMPT AND ABORT service actions. For the REGISTER and REGISTER AND IGNORE EXISTING KEY service action, the SERVICE ACTION RESERVATION KEY field contains the new reservation key to be registered. For the PREEMPT and PREEMPT AND ABORT service actions, the SERVICE ACTION RESERVATION KEY field contains the reservation key of the persistent reservations that are being preempted. The SERVICE ACTION RESERVATION KEY field is ignored for all other service actions.

If the scope is an ELEMENT_SCOPE reservation, the SCOPE-SPECIFIC ADDRESS field shall contain the element address, zero filled in the most significant bits to fit the field. If the service action is REGISTER, REGISTER AND IGNORE EXISTING KEY, or CLEAR or if the scope is a LU_SCOPE reservation, the SCOPE-SPECIFIC ADDRESS field shall be set to zero.

The Activate Persist Through Power Loss (APTPL) bit is valid only for the REGISTER and the REGISTER AND IGNORE EXISTING KEY service actions. In all other cases, the APTPL bit shall be ignored. Support for an APTPL bit equal to one is optional. If a device server that does not support the APTPL bit value of one receives that value in a REGISTER or a REGISTER AND IGNORE EXISTING KEY service action, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense code shall be set to INVALID FIELD IN PARAMETER LIST.

If the last valid APTPL bit value received by the device server is zero, the loss of power in the target shall release the persistent reservation for all logical units and remove all reservation keys (see 5.5.3.4). If the last valid APTPL bit value received by the device server is one, the logical unit shall retain any persistent reservation(s) that may be present and all reservation keys for all initiators even if power is lost and later returned (see 5.5.3.2).

The IGNORE TARGET PORT bit is valid only for the REGISTER and REGISTER AND IGNORE EXISTING KEY service actions, and shall be ignored for all other service actions. The IGNORE TARGET PORT bit ensures the application client and target port agree on the current value of the ignore target port mode flag (see 7.11.4). If the ignore target port mode is true and a device server receives a REGISTER or REGISTER AND IGNORE EXISTING KEY service action with the IGNORE TARGET PORT bit set to zero, it shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN PARAMETER LIST. If the ignore target port mode is false (see 7.11.4) and a device server receives a REGISTER or REGISTER AND IGNORE EXISTING KEY service action with the IGNORE TARGET PORT bit set to one, it shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of INVALID FIELD IN PARAMETER LIST.

Table 76 summarizes which fields are set by the application client and interpreted by the device server for each service action and scope value.

Table 76 — PERSISTENT RESERVE OUT service actions and valid parameters

		Parameters					
Service action	**Allowed SCOPE**	**TYPE**	**RESERVATION KEY**	**SERVICE ACTION RESERVATION KEY**	**SCOPE-SPECIFIC ADDRESS**	**IGNORE TARGET PORT**	**APTPL**
REGISTER	ignored	ignored	valid	valid	ignored	valid	valid
REGISTER AND IGNORE EXISTING KEY	ignored	ignored	ignored	valid	ignored	valid	valid
RESERVE	LU_SCOPE ELEMENT_SCOPE	valid valid	valid	ignored ignored	ignored valid (element)	ignored	ignored
RELEASE	LU_SCOPE ELEMENT_SCOPE	valid valid	valid	ignored ignored	ignored valid (element)	ignored	ignored
CLEAR	ignored	ignored	valid	ignored	ignored	ignored	ignored
PREEMPT	LU_SCOPE ELEMENT_SCOPE	valid valid	valid	valid valid	ignored valid (element)	ignored	ignored
PREEMPT & ABORT	LU_SCOPE ELEMENT_SCOPE	valid valid	valid	valid valid	ignored valid (element)	ignored	ignored

7.11.4 Set Capabilities

The SET CAPABILITIES service action requests that the device server enable or disable support for various persistent reservation features.

Table 73. SET CAPABILITIES parameter data

Byte\Bit	7	6	5	4	3	2	1	0
0	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	Rsvd	~~Rsvd~~ APPLICATION NAME MODE	IGNORE TARGET PORT MODE
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	Reserved							

If the IGNORE TARGET PORT MODE bit is supported, the target port shall maintain an ignore target port mode flag. An ignore target port mode flag of true indicates the target port shall not include the relative target port identifier in the initiator identifying information (see 5.5.3.x). An ignore target port mode flag of false indicates the target port shall include the relative target port identifier in the initiator identifying information. The ignore target port mode flag shall be preserved through power loss.

An IGNORE TARGET PORT MODE bit of one indicates that the target port shall set its ignore target port mode flag to true. An IGNORE TARGET PORT MODE bit of zero indicates that the target port shall set its ignore target port mode flag to false.

If a change in the ignore target port mode flag is requested when any registrations or reservations are being stored in the logical unit, the device server shall create a CHECK CONDITION with a sense code of ILLEGAL REQUEST and an additional sense code of REGISTRATIONS PRESENT.

If the APPLICATION NAME MODE bit is supported, the device server shall maintain an application name mode flag. An application name mode flag of true indicates the application client shall set the USE APPLICATION NAME bit to true in the CDB of every PERSISTENT RESERVE OUT command and provide an application name in the APPLICATION NAME field in the parameter data. An application name mode flag of false indicates the application client shall not set the USE APPLICATION NAME bit to true in the CDB of any PERSISTENT RESERVE OUT command. The application name mode flag shall be preserved through power loss.

7.11.5 Association service actions parameter list

The parameter list required to perform the PERSISTENT RESERVE OUT command with the ASSOCIATE APPLICATION NAME and UNASSOCIATE APPLICATION NAME service actions is defined in table 755.

Table 755. ASSOCIATE APPLICATION NAME and UNASSOCIATE APPLICATION NAME parameter data

Byte\Bit	7	6	5	4	3	2	1	0
0	APPLICATION NAME							
15								
16	Reserved							
..								
39								

The APPLICATION NAME field is a value provided by the application client to identify the application.

For an ASSOCIATE APPLICATION NAME service action, the device server shall associate the initiator port identifier of the initiator port used to send the PERSISTENT RESERVE OUT command with the specified application name. Any previous association for that initiator port identifier shall be replaced. Each initiator port is associated with at most one application name.

For an UNASSOCIATE APPLICATION NAME service action, the device server shall remove the association of the initiator port identifier of the initiator port used to send the PERSISTENT RESERVE OUT command with the specified application name. If no such association exists, the device server shall return a CHECK CONDITION status with a sense key of ILLEGAL REQUEST and an additional sense code of NO APPLICATION NAME ASSOCIATION.

[Editor’s note: assign additional sense code for ILLEGAL REQUEST/NO APPLICATION NAME ASSOCIATION]