03-232r5 Modified Reservation Handling proposal

Date: January 14, 2004

To: T10 Committee (SCSI)

From: Roger Cummings (VERITAS)

Subject: T10/03-232r5 SPC-3 Proposal for Modified Reservation Handling

Revision History

03-232r0 (July 2, 2003) First Revision
03-232r1 (November 4, 2003) Added new rows and footnotes to Table 31 rather than a new subclause to the Reservations model subclause.
03-232r2 (November 7, 2003) Moved text from footnotes to new subclause.
03-232r3 (December 30, 2003) Incorporated suggested clarifications in new subclause, renamed bit to differentiate from other capabilities.
03-232r4 (January 12, 2004) Incorporated editorial comments from George Penokie.
03-232r5 (January 14, 2003) Incorportaed chnages from January 2004 CAP meeting.

Related Documents

02-483r0 (November 6, 2002) Further Work on Persistent Reservations. First problem statement following on SPC-2 work.
02-483r1 (May 5, 2003) Two Persistent Reservations problems - updates & decision required. Details of current usage and suggested new features and their usage explored in detail.
03-231r0 (July 2, 2003) Two Persistent Reservations problems - latest information. Feedback on earlier approaches and results from latest testing & investigation.
spc3r15 – SCSI Primary Commands – 3 revision 15

Overview

Reserves and Releases are presently used extensively in situations where sequential-access devices are employed. However because there is no way of checking for the existence of a reservation, multiple reserves and releases are known to be issued against the same device by applications, device drivers and other entities that coexist in a computer system. When combined with the definitions contained in SPC-2, this situation makes the deployment of persistent reservations very problematic, because it would require the application, platform and other entities to be upgraded simultaneously, and that is unacceptable for a number of both technical and business reasons. A number of approaches to relieving this situation have been investigated during the last year, all of which have provided some scheme for allowing mixed usage of reservations and persistent reservations in accessing a storage device. This proposal treats reserve and release as NOPs if they are generated by a computer system that already has access to the device through an existing persistent reservation. In all other cases, the behavior is as defined in SPC-2.

Proposal

This document proposes a modification to the method of handling RESERVE and RELEASE commands in the presence of a Persistent Reservation that was defined in SPC-2. The modification is proposed for inclusion in SPC-3 and is NOT backwards compatible. This proposal is made to address the need to be able to mix usage of reserve/release and persistent reservations when dealing with tapes as identified in 02-483r1 and further addressed in 03-231r0. An addition to the REPORT CAPBILITIES service action is proposed to indicate that this modified reservation handling is supported. The modification is not selectable.

The proposal consists of three parts:

1) Add two rows to Table 31.

2) Add a new subclause (5.6.2) to contain the exceptions to SPC-2 RESERVE AND RELEASE bahavior.
3) Add a bit to the PERSISTENT RESERVE IN parameter data for the REPORT CAPABILITIES service action in subclause 6.11.1

Suggested Changes

5.6.1 Reservations overview

Reservations may be used to allow a device server to execute commands from a selected set of I_T nexuses (i.e., combinations of initiator ports accessing target ports) and reject commands from I_T nexuses outside the selected set. The device server uniquely identifies I_T nexuses using protocol specific mechanisms.

Application clients may add or remove I_T nexuses from the selected set using reservation commands. If the application clients do not cooperate in the reservation protocol, data may be unexpectedly modified and deadlock conditions may occur.

The scope of a reservation shall be one of the following:

a) **Logical unit reservations** - a logical unit reservation restricts access to the entire logical unit; and
b) **Element reservations** - an element reservation restricts access to a specified element within a medium changer.

Reservations may be further qualified by restrictions on types of access (e.g., read, write). However, any restrictions based on the type of reservation are independent of the scope of the reservation.

Reservation restrictions are placed on commands as a result of access qualifiers associated with the type of reservation. The details of which commands are allowed under what types of reservations are described in table 31. If any element is reserved within a logical unit, that logical unit shall be considered reserved for the commands listed in table 31 and the allowed/conflict information in table 31 shall apply.

In table 31 and table 32 the following key words are used:

allowed: Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present should complete normally.

conflict: Commands received from I_T nexuses not holding the reservation or from I_T nexuses not registered when a registrants only or all registrants type persistent reservation is present shall not be performed and the device server shall terminate the command with a RESERVATION CONFLICT status.

Commands from I_T nexuses holding a reservation should complete normally. The behavior of commands from registered I_T nexuses when a registrants only or all registrants persistent reservation is present is specified in table 31 and table 32.

An unlinked command shall be checked for reservation conflicts before the task containing that command enters the enabled task state. The reservation state as it exists when the first command in a group of linked commands enters the enabled task state shall be used in checking for reservation conflicts for all the commands in the task. Once a task has entered the enabled task state, the command or commands comprising that task shall not be terminated with a RESERVATION CONFLICT due to a subsequent reservation. Any command in a group of linked commands that changes the reservation state shall be the last command in the group.

For each command, this standard or a related command standard (see 3.1.17) defines the conditions that result in RESERVATION CONFLICT. Command standards define the conditions either in the device model (preferred) or in the descriptions each specific command.

Table 31 — SPC commands that are allowed in the presence of various reservations (part 1 of 2)

Command	Addressed LU has this type of persistent reservation held by another I_T nexus				
	From any I_T nexus		From registered I_T nexus (RR all types)	From I_T nexus not registered	
	Write Excl	Excl Access		Write Excl RR	Excl Access – RR
ACCESS CONTROL IN	Allowed	Allowed	Allowed	Allowed	Allowed
ACCESS CONTROL OUT	Allowed	Allowed	Allowed	Allowed	Allowed
CHANGE ALIASES	Conflict	Conflict	Allowed	Conflict	Conflict
EXTENDED COPY	Conflict	Conflict	Allowed	Conflict	Conflict
INQUIRY	Allowed	Allowed	Allowed	Allowed	Allowed
LOG SELECT	Conflict	Conflict	Allowed	Conflict	Conflict
LOG SENSE	Allowed	Allowed	Allowed	Allowed	Allowed
MODE SELECT(6)/ MODE SELECT(10)	Conflict	Conflict	Allowed	Conflict	Conflict
MODE SENSE(6)/ MODE SENSE(10)	Conflict	Conflict	Allowed	Conflict	Conflict
PERSISTENT RESERVE IN	Allowed	Allowed	Allowed	Allowed	Allowed
PERSISTENT RESERVE OUT	see table 32				
PREVENT ALLOW MEDIUM REMOVAL (Prevent=0)	Allowed	Allowed	Allowed	Allowed	Allowed
PREVENT ALLOW MEDIUM REMOVAL (Prevent<>0)	Conflict	Conflict	Allowed	Conflict	Conflict
READ ATTRIBUTE	Conflict	Conflict	Allowed	Conflict	Conflict
READ BUFFER	Conflict	Conflict	Allowed	Conflict	Conflict
READ MEDIA SERIAL NUMBER	Allowed	Allowed	Allowed	Allowed	Allowed
RECEIVE COPY RESULTS	Conflict	Conflict	Allowed	Conflict	Conflict
RECEIVE DIAGNOSTIC RESULTS	Conflict	Conflict	Allowed	Conflict	Conflict
RELEASE	As defined in SPC-2 [a]				
REPORT ALIASES	Allowed	Allowed	Allowed	Allowed	Allowed
REPORT DEVICE IDENTIFIER	Allowed	Allowed	Allowed	Allowed	Allowed
REPORT LUNS	Allowed	Allowed	Allowed	Allowed	Allowed

Key: **LU**=Logical Unit, **Excl**=Exclusive, **RR**=Registrants Only or All Registrants, **<>** Not Equal

[a] Exceptions to the behavior of the RESERVE and RELEASE commands described in SPC-2 are defined in 5.6.2.

Table 31 — SPC commands that are allowed in the presence of various reservations (part 2 of 2)

Command	Addressed LU has this type of persistent reservation held by another I_T nexus				
	From any I_T nexus		From registered I_T nexus (RR all types)	From I_T nexus not registered	
	Write Excl	Excl Access		Write Excl RR	Excl Access – RR
REPORT SUPPORTED OPERATION CODES	Conflict	Conflict	Allowed	Conflict	Conflict
REPORT SUPPORTED TASK MANAGEMENT FUNCTIONS	Conflict	Conflict	Allowed	Conflict	Conflict
REPORT TARGET PORT GROUPS	Allowed	Allowed	Allowed	Allowed	Allowed
REQUEST SENSE	Allowed	Allowed	Allowed	Allowed	Allowed
RESERVE	As defined in SPC-2 [a]				
SEND DIAGNOSTICS	Conflict	Conflict	Allowed	Conflict	Conflict
SET DEVICE IDENTIFIER	Conflict	Conflict	Allowed	Conflict	Conflict
SET TARGET PORT GROUPS	Conflict	Conflict	Allowed	Conflict	Conflict
TEST UNIT READY	Allowed	Allowed	Allowed	Allowed	Allowed
WRITE ATTRIBUTE	Conflict	Conflict	Allowed	Conflict	Conflict
WRITE BUFFER	Conflict	Conflict	Allowed	Conflict	Conflict

Key: **LU**=Logical Unit, **Excl**=Exclusive, **RR**=Registrants Only or All Registrants, **<>** Not Equal

[a] Exceptions to the behavior of the RESERVE and RELEASE commands described in SPC-2 are defined in 5.6.2.

Table 32 — PERSISTENT RESERVE OUT service actions that are allowed in the presence of various reservations

Command Service Action	Addressed LU has a persistent reservation held by another I_T nexus	
	Command is from a registered I_T nexus	Command is from a not registered I_T nexus
CLEAR	Allowed	Conflict
PREEMPT	Allowed	Conflict
PREEMPT & ABORT	Allowed	Conflict
REGISTER	Allowed	Allowed
REGISTER AND IGNORE EXISTING KEY	Allowed	Allowed
RELEASE	Allowed [a]	Conflict
RESERVE	Conflict	Conflict

Key: **LU**=Logical Unit

[a] The reservation is not released (see 5.6.2.7.2).

The time at which a reservation is established with respect to other tasks being managed by the device server is vendor specific. Successful completion of a reservation command indicates that the new reservation is established. A reservation may apply to some or all of the tasks in the task set before the completion of the reservation command. The reservation shall apply to all tasks received by the device server after successful completion of the reservation command. Any persistent reserve service action shall be performed as a single indivisible event.

Multiple persistent reserve service actions may be present in the task set at the same time. The order of execution of such service actions is defined by the tagged queuing restrictions, if any, but each is executed as a single indivisible command without any interleaving of actions that may be required by other reservation commands.

5.6.2 Exceptions to SPC-2 RESERVE and RELEASE behavior

This subclause defines exceptions to the behavior of the RESERVE and RELEASE commands defined in SPC-2. The RESERVE and RELEASE commands are obsolete in this standard, except for the behavior defined in this subclause. Device Servers that operate using the exceptions described in this subclause shall set the CRH bit of the parameter data for the REPORT CAPABILITIES service action of the PERSISTENT RESERVE IN command to one.

A RELEASE command shall complete, but the persistent reservation shall not be released, when the RELEASE command is received from:

a) An I_T nexus that is a persistent reservation holder (see 5.6.2.6), or;
b) An I_T nexus that is registered if a registrants only or all registrants type persistent reservation is present.

A RESERVE command shall complete, but no reservation shall be established and the persistent reservation shall not be changed, if the RESERVE command is received from:

a) An I_T nexus that is a persistent reservation holder (see 5.6.2.6), or;
b) An I_T nexus that is registered if a registrants only or all registrants type persistent reservation is present.

In all other cases, a RESERVE or a RELEASE command shall not be processed, and the device server shall terminate the command with a RESERVATION CONFLICT status, as defined in SPC-2.

6.11. PERSISTENT RESERVE IN command

[6.11.1 PERSISTENT RESERVE IN command introduction]

[6.11.2 PERSISTENT RESERVE IN service actions]

[6.11.3 PERSISTENT RESERVE IN parameter data for READ KEYS]

[6.11.4 PERSISTENT RESERVE IN parameter data for READ RESERVATION]

6.11.5 PERSISTENT RESERVE IN parameter data for REPORT CAPABILITIES

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the REPORT CAPABILITIES service action is shown in table 103.

Table 103 — PERSISTENT RESERVE IN parameter data for REPORT CAPABILITIES

Bit / Byte	7	6	5	4	3	2	1	0
0	(MSB)							
1				LENGTH (0008h)				(LSB)
2		Reserved		CRH	SIP_C	ATP_C	ES_C	PTPL_C
3	TMV			Reserved				PTPL_A
4								
5				PERSISTENT RESERVATION TYPE MASK				
6								
7				Reserved				

The LENGTH field indicates 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.

A CRH (Compatible Reservation Handling) bit set to one indicates that the device server supports the exceptions to the SPC-2 RESERVE and RELEASE commands described in 5.6.2. A CRH bit set to zero indicates that RESERVE and RELEASE commands are processed as defined in SPC-2.

An SIP_C (Specify Initiator Ports Capable) bit set to one indicates that the device server supports the SPEC_I_PT bit in the PERSISTENT RESERVE OUT command parameter data (see 6.12.3). An SIP_C bit set to zero indicates that the device server does not support the SPEC_I_PT bit in the PERSISTENT RESERVE OUT command parameter data.

An ATP_C (All Target Ports Capable) bit set to one indicates that the device server supports the ALL_TG_PT bit in the PERSISTENT RESERVE OUT command parameter data. An ATP_C bit set to zero indicates that the device server does not support the ALL_TG_PT bit in the PERSISTENT RESERVE OUT command parameter data.

An PTPL_C (Persist Through Power Loss Capable) bit set to one indicates that the device server supports the persist through power loss capability (see 5.6.2.2) for persistent reservations and the APTPL bit in the PERSISTENT RESERVE OUT command parameter data. An PTPL_C bit set to zero indicates that the device server does not support the persist through power loss capability.

An ES_C (Element Scope Capable) bit set to one indicates that the device server supports a SCOPE value of ELEMENT_SCOPE (see 6.11.4.2) in PERSISTENT RESERVE OUT commands (see 6.12). An ES_C bit set to zero indicates that the device server does not support a SCOPE value of ELEMENT_SCOPE in PERSISTENT RESERVE OUT commands.

A TMV (Type Mask Valid) bit set to one indicates the PERSISTENT RESERVATION TYPE MASK field contains a bit map indicating which persistent reservation types are supported by the device server. A TMV bit set to zero indicates the PERSISTENT RESERVATION TYPE MASK field shall be ignored.

A PTPL_A (Persist Through Power Loss Activated) bit set to one indicates that persist through power loss capability (see 5.6.2.2) is activated because the most recent successfully completed PERSISTENT RESERVE OUT command with REGISTER or REGISTER AND IGNORE EXISTING KEY service action had the APTPL bit set to one in the parameter data. A PTPL_A bit set to zero indicates that the persist through power loss capability is not activated.

The PERSISTENT RESERVATION TYPE MASK field (see table 104) contains a bit map that indicates the persistent reservation types that are supported by the device server.

Table 104 — Persistent Reservation Type Mask format

Bit / Byte	7	6	5	4	3	2	1	0
4	WR_EX_AR	EX_AC_RO	WR_EX_RO	Reserved	EX_AC	Reserved	WR_EX	Reserved
5	Reserved							EX_AC_AR

A WR_EX_AR (Write Exclusive – All Registrants) bit set to one indicates that the device server supports the Write Exclusive – All Registrants persistent reservation type. An WR_EX_AR bit set to zero indicates that the device server does not support the Write Exclusive – All Registrants persistent reservation type.

An EX_AC_RO (Exclusive Access – Registrants Only) bit set to one indicates that the device server supports the Exclusive Access – Registrants Only persistent reservation type. An EX_AC_RO bit set to zero indicates that the device server does not support the Exclusive Access – Registrants Only persistent reservation type.

A WR_EX_RO (Write Exclusive – Registrants Only) bit set to one indicates that the device server supports the Write Exclusive – Registrants Only persistent reservation type. An WR_EX_RO bit set to zero indicates that the device server does not support the Write Exclusive – Registrants Only persistent reservation type.

An EX_AC (Exclusive Access) bit set to one indicates that the device server supports the Exclusive Access persistent reservation type. An EX_AC bit set to zero indicates that the device server does not support the Exclusive Access persistent reservation type.

An WR_EX (Write Exclusive) bit set to one indicates that the device server supports the Write Exclusive persistent reservation type. An WR_EX bit set to zero indicates that the device server does not support the Write Exclusive persistent reservation type.

An EX_AC_AR (Exclusive Access – All Registrants) bit set to one indicates that the device server supports the Exclusive Access – All Registrants persistent reservation type. An EX_AC_AR bit set to zero indicates that the device server does not support the Exclusive Access – All Registrants persistent reservation type.