Date: July 03, 2003

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: SPC-3: Persistent All Registrants Fix

1 Overview Problem 1

The following question was received on the T10 reflector from Ken Craig:

SPC-3 Rev. 13, Figure 3 (the PREEMPT flow chart) now shows that when a LUN has a reservation type of ALL REGISTRANTS and gets one of the PREEMPT Service Actions with a Service Action Reservation Key of 0 all of the registrations except the one that belongs to the Initiator that sent the PREEMPT are removed. However the third paragraph of Section 5.5.2.7.1.2 states that all of the LUN's registered Initiators whose reservation key was not removed get a UA with one of the ASCs described in the text below the paragraph. It seems like item a) in that text can't apply in this case since all of the registered Initiators have had their reservation removed except for the Initiator that sent the PREEMPT who never gets UA. Is there a conflict between Section 5.5.2.7.1.2 and the revised flow chart or am I interpreting this incorrectly?

2 Response

The wording you pointed out is not in conflict with figure 3 but it is weird. You are correct in that there are no registered initiators left to create a unit attention for if the key is set to zero. The wording states that any registrations that are left get the UA but there are none left so therefore there are no UAs.

That said, the wording is meaningless and could only lead to confusion so it should be removed.

3 Proposal

The following is to be removed from section 5.5.2.7.1.2 Handling for released all registrants persistent reservations in the next revision of SPC-3:

If a persistent reservation was removed or changed, the device server shall establish a unit attention for every initiator port associated with a registered I_T nexus whose reservation key was not removed except for the initiator port through which the command was issued. The additional sense code shall be set as follows:

a) If the service action was PREEMPT or PREEMPT AND ABORT with a SERVICE ACTION RESERVATION KEY set to zero, the additional sense code shall be set to RESERVATIONS RELEASED.
b) If the service action was RELEASE, the additional sense code shall be set to RESERVATIONS RELEASED.

4 Overview Problem 2

There is no way to tell which types of persistent reservation a logical unit supports other than just trying it. If the logical unit does not support it a check condition should occur for that command. But that is not a very good way to find out.

5 Solution

Add a two sets of bits into the REPORT CAPABILITIES service actions parameter data to which indicates which persistent reservation types are supported.

One set would be four bits in size for the original remaining four types and the other would be two bits in size for the two new types (i.e., ALL REGISTRANTS types).

The bits for the original four types will be defined as set to zero for logical units that support the type and to one for logical units that do not support the type. The all new types (including all registrants) will be defined as set to one for logical units that support the type and to zero for logical units that do not support the type.

This is necessary to allow compatibility with existing implementations.

6 Proposal

Change section 6.11.5 PERSISTENT RESERVE IN parameter data for REPORT CAPABILITIES to the following:

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 1.

Table 1 — PERSISTENT RESERVE IN parameter data for REPORT CAPABILITIES

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

An TYPE_MASK (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 TYPE_MASK bit set to zero indicates the PERSISTENT RESERVATION TYPE MASK field shall be ignored.

The PERSISTENT RESERVATION TYPE MASK field (table 2) contains a bit map of the persistent reservation types that are supported by the device server.

Table 2 — PERSISTENT RESERVATION TYPE MASK **field**

Bit Byte	7	6	5	4	3	2	1	0
0	WR_EX_ALL	EX_ACC_RO	WR_EX_RO	Reserved	EX_ACC	Reserved	WR_EX	Reserved
1	Reserved							EX_ACC_ALL

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

An 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 bit set to zero indicates that the device server does not support the write exclusive-registrants only persistent reservation type.

An EX_ACC_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_ACC bit set to zero indicates that the device server does not support the exclusive access-registrants only persistent reservation type.

An WR_EX_ALL (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_ALL bit set to zero indicates that the device server does not support the write exclusive-all registrants persistent reservation type.

An EX_ACC_ALL (Exclusive Access-All Registrants) bit set to one indicates that the device server supports the exclusive access-all registrants persistent reservation type. An EX_ACC_ALL bit set to zero indicates that the device server does not support the exclusive access-all registrants persistent reservation type.

7 Overview Problem 3

It is not clear what should happen if an All Registrants reservation is in effect, an ACA is in effect, and an initiator other than the faulted initiator issues a Preempt and Abort service action.

The answer it that question is: The logical unit should to return an ACA ACTIVE or a BUSY status.

That is not clear in the standard and wording will be proposed to fix that.

8 Solution

The logical unit should to return an ACA ACTIVE or a BUSY status depending on the TST bit setting.

9 Proposal

Change the following from section 5.5.2.7.5 Preempting and aborting:

The application client’s request for and the device server’s responses to a PERSISTENT RESERVE OUT command PREEMPT AND ABORT service action are identical to the responses to a PREEMPT service action (see 5.5.2.7.4) except for the following additions. If no reservation conflict occurred, the device server shall perform the following uninterrupted series of actions:

a) If the TST field is 000b (see 7.4.6) and ACA condition exists for initiator ports other than the initiator port associated with the I_T nexus that is being preempted, the PERSISTENT RESERVE OUT command shall be terminated prior to processing with a status of ACA ACTIVE if the NACA bit equals one in the CDB CONTROL byte (see SAM-2) or BUSY if the NACA equals zero. If the TST field contains 001b, then ACA condition for initiator ports other than the initiator port associated with the I_T nexuses that are being preempted shall not prevent the processing of the PERSISTENT RESERVE OUT command;

to:

The application client’s request for and the device server’s responses to a PERSISTENT RESERVE OUT command PREEMPT AND ABORT service action are identical to the responses to a PREEMPT service action (see 5.5.2.7.4) except for the following additions. If no reservation conflict occurred, the device server shall perform the following uninterrupted series of actions:

a) If the TST field contains 001b, then an ACA condition for initiator ports other than the initiator port associated with the I_T nexuses that are being preempted shall not prevent the processing of the PERSISTENT RESERVE OUT command. If the TST field is 000b (see 7.4.6) the PERSISTENT RESERVE OUT command shall be terminated prior to processing with a status of ACA ACTIVE if the NACA bit equals one in the CDB CONTROL byte (see SAM-2) or BUSY if the NACA equals zero if one of the following conditions exists:
   A) An ACA condition exists for initiator ports other than the initiator port associated with the I_T nexus that is being preempted; or

B) An ACA condition exists and the reservation being preempted is an All Registrants type reservation;