

T10/03-230 revision 1

Date: July 02, 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) _____							
1	LENGTH (0008h)							(LSB)
2	Reserved			SIP_C	ATP_C	ES_C	PTPL_C	
3	Reserved						PTPL_A	
4	Reserved			EX_ACC_RO	WR_EX_RO	EX_ACC	WR_EX	
5	Reserved					EX_ACC_ALL	WR_EX_ALL	
6	Reserved							
7	Reserved							

An `WR_EX` (Write Exclusive) bit of one indicates that the device server does not support the write exclusive persistent reservation type. An `WR_EX` bit of zero indicates that the device server supports the write exclusive persistent reservation type.

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

An `WR_EX_RO` (Write Exclusive-Registrants Only) bit of one indicates that the device server does not support the write exclusive-registrants only persistent reservation type. An `WR_EX` bit of zero indicates that the device server supports the write exclusive-registrants only persistent reservation type.

An `EX_ACC_RO` (Exclusive Access-Registrants Only) bit of one indicates that the device server does not support the exclusive access-registrants only persistent reservation type. An `EX_ACC` bit of zero indicates that the device server supports the exclusive access-registrants only persistent reservation type.

An `WR_EX_ALL` (Write Exclusive-All Registrants) bit of one indicates that the device server supports the write exclusive-all registrants persistent reservation type. An `WR_EX_ALL` bit of 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 of one indicates that the device server supports the exclusive access-all registrants persistent reservation type. An `EX_ACC_ALL` bit of zero indicates that the device server does not support the exclusive access-all registrants persistent reservation type.