

To: T10 Technical Committee  
 From: Rob Elliott, Compaq Computer Corporation (Robert.Elliott@compaq.com)  
 Date: 7 March 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.

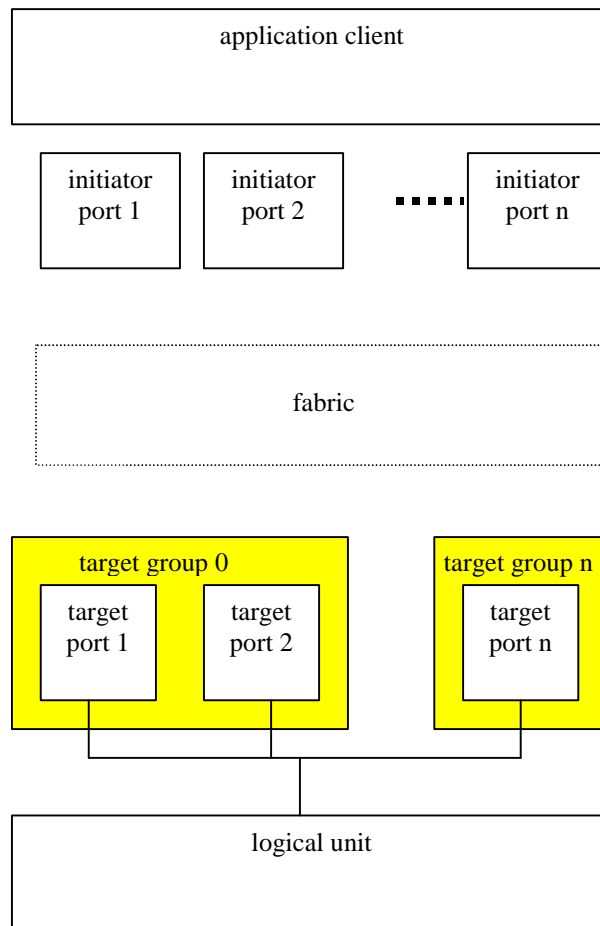
### **Related Documents**

T10/spc2r19 – SCSI Primary Commands revision 19 (by Ralph Weber)

T10/00-232r6 – Asymmetric target behavior (by Ken Moe)

### **Overview**

SPC-2 requires that logical units remember the initiator port through which a reservation was made in addition. An application client 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 application clients. Requiring reservation commands for each initiator port burdens the application client 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 one cluster.

In some protocols (SRP and iSCSI) the initiator port definition is at a high level and the problems can be minimized. In Fibre Channel, however, the initiator port is the Fibre Channel port. A higher level identifier such as the Fibre Channel node or FC-GS-3 platform name would be preferable.

There is no generic term for this level of name in SAM-2. It could be defined as the initiator device identifier hinted at in SAM-2. For SRP and iSCSI, the initiator device identifier would just be the same as the initiator port identifier.

**Table 1. Initiator device identifiers**

<b>Protocol</b>	<b>initiator device identifier</b>
SPI	SCSI device ID
FCP	FC-GS-3 platform name
SRP	initiator port identifier
iSCSI	initiator port identifier

When communicating across multiple SCSI transports (e.g. InfiniBand SRP to Fibre Channel FCP to TCP/IP iSCSI), the endpoint transports might support device identifiers but the intermediate transports do not. Just adding a bit to the PERSISTENT RESERVE OUT CDB or adding a field in a mode page is insufficient.

A new command could be defined whose payload is the purported initiator device identifier as known to the application client. When the device server receives the command, it checks the initiator device identifier reported by the transport and compares it to the payload. If the identifiers match, the device server reports command success. If they do not match, the device server reports command failure and the initiator device identifier cannot be used.

However, if the application client is going to run such a command, it might as well run another command that explicitly registers the initiator device identifier with the target. This avoids the need for intermediate transports to pass through the initiator device identifier. This makes it like the AccessID used by access controls. Any initiator that has registered with this ID is granted the reservation.

Alternatively, this identifier could be passed as data in the PERSISTENT RESERVE OUT command.

This proposal allows persistent reservation commands to use the initiator device identifier rather than initiator port identifier. Application clients only enable this mode after they have tested the SCSI domain for successful initiator device identifier delivery.

### **Suggested Changes**

Text is from SPC-2 revision 19. Option a) is documented.

#### **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 ~~on a specific port of a device server~~. 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.

...

### 5.5.3.2 Preserving persistent reservations

...

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 identifier;  
[aa\) relative port identifier of the target port;](#)
- b) reservation key; and
- c) when supported by the protocol, the initiator port's world wide identification.

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 identifier;  
[aa\) relative port identifier of the target port;](#)
- b) reservation key;
- c) scope;
- d) type; and
- e) when supported by the protocol, the initiator port's world wide identification.

For those protocols for which the initiator port's world wide identification is available to the device server the initiator port's world wide identification shall be used to determine if the initiator identifier has changed. This determination shall be made at any time the target detects that the configuration of the system may have changed. If the initiator identifier changed, the device server shall assign the new initiator identifier to the existing registration and reservation of the initiator port having the same world wide identification.

...

### 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 identification and, if available, the initiator port's world wide identification;
- 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 service delivery ports provide alternate service delivery 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 device has more than one service delivery port, the arbitration and connection management among the service delivery ports is vendor specific. If one service delivery port is being used by an initiator, accesses attempted through other service delivery port(s) may:

- a) receive a status of BUSY; or
- b) be accepted as if the other service delivery port(s) were not in use.

The device server shall indicate the presence of multiple 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 on the same service delivery port except the initiator holding the reservation and all initiators on all other service delivery ports. 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 logical units in the target and for all initiators (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 the selected logical unit for all initiators. Most other logical unit states remain unmodified, including MODE SELECT parameters, reservations, and ACA (see SAM-2).

### 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	<a href="#">USE INITIATOR DEVICE IDENTIFIER</a> Rsvd	Rsvd	SERVICE ACTION				
2	SCOPE				TYPE			
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	PARAMETER LIST LENGTH (18h)							
8								
15	CONTROL							

...

**Table 73. CHECK TRANSPORT INITIATOR DEVICE IDENTIFIER command**

Byte\Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (5Fh)							
1	Rsvd	<a href="#">USE INITIATOR DEVICE IDENTIFIER Rsvd</a>	Rsvd	SERVICE ACTION				
2	SCOPE				TYPE			
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	PARAMETER LIST LENGTH (18h)							
8								
15	CONTROL							

...