June 22, 1999

To: T10 Committee

From: Tom Coughlan and John Meneghini
Compaq Computer Corporation
110 Spitbrook Road
Nashua, New Hampshire
Telephone: 603-884-0933
603-884-3102
E-mail: tom.coughlan@compaq.com
john.meneghini@compaq.com

Subject: Persistent Reservation: Register and Ignore Existing Key (99-199r0)

The current revision of SPC-2 requires that an initiator with a previously established reservation key must know the key value before it can perform a Register service action. There are situations, described below, where it is unnecessary for the initiator to determine its old key before re-registering. This proposal creates a new Persistent Reserve Out service action that allows an initiator to register without knowing the existing key value.

The current Reservation behavior is designed to encourage an initiator that has exited a multi-initiator environment, and has lost memory of its key, to reestablish communication with the other initiators before it changes the reservation state. This provides a minimal level of protection against an initiator that has rebooted and might inadvertently change the reservation state before it has coordinated with other initiators who are sharing the device. The protection is minimal, because:

- A re-joining initiator is free to do I/O under the auspices of the existing registration and reservation. It is only blocked from changing the reservation state.
- A new initiator, possibly running a different multi-initiator sharing protocol, is free to register with any key, change the reservation state, and do I/O within the limits allowed by the current reservation state. This initiator will not encounter any reservation error at all.
- A malicious initiator can read the existing keys from the device server and attempt to register with each one until it finds its old key.

Considering these limitations, there are some multi-initiator environments where the overhead and delay associated with communicating old keys to re-joining initiators is not considered worth the benefit. These environments rely on slightly more trust, in an already highly trusted environment, in order to reduce the need for inter-system communication and complex failure recovery algorithms, especially during booting.

The proposed change is to create a new service action called "Register, and Ignore Existing Key". This action is similar to the Register service action, except that the Reservation Key field is ignored by the device server.

Detailed Change:

The following changes are proposed to SPC-2 (Rev. 9).

3.1.40 registered: The condition that exists for an initiator (or application client) from successful completion of a PRESISTENT RESERVE OUT command with a REGISTER or REGISTER and IGNORE EXISTING KEY service action (see 5.3.2.3) until the initiator registration is removed (see 5.3.2.5). Note: the term REGISTER is used in some places to refer to both the REGISTER and the REGISTER and IGNORE EXISTING KEY service actions.

5.3 Reservations

The two methods are prevented from creating conflicting and undefined interactions using RESERVATION CONFLICT status in the following manner. If a logical unit has executed a PERSISTENT RESERVE OUT command with ~~the~~ a REGISTER service action and is still registered by any initiator, all RESERVE commands and all RELEASE commands regardless of initiator shall conflict and shall terminate with a RESERVATION CONFLICT status. If a logical unit has been reserved by any RESERVE command and is still reserved by any initiator, all PERSISTENT RESERVE IN and all PERSISTENT RESERVE OUT commands shall conflict regardless of initiator or service action and shall terminate with a RESERVATION CONFLICT status.

Table 7. Add a row for the REGISTER and IGNORE EXISTING KEY command. This row has the same data as the row for the REGISTER command.

5.3.2.1 Preserving persistent reservations

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

5.3.2.3 Registering

To establish a persistent reservation the initiator shall first register with a logical unit.
If the initiator has not yet established a reservation key or the reservation key has been removed, the registration is accomplished by issuing a PERSISTENT RESERVE OUT command with service action of REGISTER with the following parameters:

a) APTPL bit optionally set to one;
b) reservation key set to zero; and
c) service action reservation key set to a non-zero value.

If the initiator has an established registration it may change its reservation key. This is accomplished by issuing a PERSISTENT RESERVE OUT command with service action of REGISTER with the following parameters:

a) APTPL bit optionally set to one;
b) reservation key set to the value of the previously established reservation key; and
c) service action reservation key set to a non-zero value.

Alternatively, a new registration can be established by issuing a PERSISTENT RESERVE OUT command with a service action of REGISTER and IGNORE EXISTING KEY and the following parameters:

a) APTPL bit optionally set to one; and
b) service action reservation key set to a non-zero value.

If a PERSISTENT RESERVE OUT with a REGISTER and IGNORE EXISTING KEY service action is sent when an established registration key exists, the registration will be superceded with the specified service action reservation key.

If a PERSISTENT RESERVE OUT with a REGISTER and IGNORE EXISTING KEY service action is sent when there is no established registration key, a new registration will be established.

If a PERSISTENT RESERVE OUT with a REGISTER service action is attempted, but there are insufficient device server resources to complete the operation, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to INSUFFICIENT REGISTRATION RESOURCES.

NOTE 3 It is recommended a target have enough resources to handle a registration from each initiator known to the target.

In response to a PERSISTENT RESERVE OUT with a REGISTER 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.

After the registration request has been processed, the device server shall then allow other PERSISTENT RESERVE OUT commands from the registered initiator to execute. For each initiator that performs a PERSISTENT RESERVE OUT with a REGISTER service action, the device server shall retain the reservation key until the key is changed by a new PERSISTENT RESERVE OUT command with ~~the~~ a REGISTER service action from the same initiator or until the initiator registration is removed (see 5.3.2.5).

Any PERSISTENT RESERVE OUT command service action received from an unregistered initiator, other than ~~the~~ a REGISTER service action, shall be rejected with a RESERVATION CONFLICT status.

5.3.2.5 Removing registrations and persistent reservations

A registered initiator using the value of the initiator/logical unit pair's established reservation key may remove a persistent reservation by issuing one of the following commands:

a) a PERSISTENT RESERVE OUT command with a service action of RELEASE from the initiator that performed the reservation (see 5.3.2.5.1);
b) a PERSISTENT RESERVE OUT command with a PREEMPT service action specifying the reservation key of the initiator holding the reservation (see 5.3.2.5.2);
c) a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action specifying the reservation key of the initiator holding the reservation (see 5.3.2.5.3); or
d) a PERSISTENT RESERVE OUT command with a service action of CLEAR service action (see 5.3.2.5.4).

A registered initiator using the value of the initiator/logical unit pair's established reservation key may remove a registration by issuing one of the following commands:

a) a PERSISTENT RESERVE OUT command with a PREEMPT service action specifying that reservation key (see 5.3.2.5.2);
b) a PERSISTENT RESERVE OUT command with a PREEMPT AND ABORT service action specifying that reservation key (see 5.3.2.5.3);
c) a PERSISTENT RESERVE OUT command with a CLEAR service action (see 5.3.2.5.4); or
d) a PERSISTENT RESERVE OUT command with a REGISTER or REGISTER and IGNORE EXISTING KEY service action from the same initiator with the value of the SERVICE ACTION RESERVATION KEY field set to zero.

7.11.2 PERSISTENT RESERVE IN parameter data for READ KEYS

The GENERATION field shall contain a 32-bit counter maintained by the device server that shall be incremented every time a PERSISTENT RESERVE OUT command requests a REGISTER, a REGISTER and IGNORE EXISTING KEY,a CLEAR, a PREEMPT, or a PREEMPT AND ABORT service action. The counter shall not be incremented by a PERSISTENT RESERVE IN command, by a PERSISTENT RESERVE OUT command that performs a RESERVE or RELEASE service action, or by a PERSISTENT RESERVE OUT command that is not performed due to an error or reservation conflict. Regardless of the APTPL bit value the generation value shall be set to 0 as part of the power on reset process.

7.12.1 PERSISTENT RESERVE OUT Service Actions

When processing the PERSISTENT RESERVE OUT service actions, the device server shall increment the generation value as specified in 7.11.2.

T10/1236-D revision 9 16 March 1999
The PERSISTENT RESERVE OUT command service actions are defined in table 48.
The parameter list values for each service action are specified in 7.12.2.

Table 48 — PERSISTENT RESERVE OUT Service action codes Code Name Description

Ed. Note: add the following to this table.

06h REGISTER and IGNORE EXISTING KEY Registers a reservation key with the device server (for more information seen 5.3.2.3).

7.12.2 PERSISTENT RESERVE OUT parameter list

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 RESERVATION KEY field in a PERSISTENT RESERVE OUT command matches the registered reservation key for the initiator from which the task was received, except for the REGISTER and IGNORE EXISTING KEY service action, where the RESERVATION KEY field is ignored, and the REGISTER service action for an unregistered initiator which shall have a reservation key value of zero. If 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. The reservation key of the initiator shall be verified to be correct regardless of the SERVICE ACTION and SCOPE field values. The obsolete field in Bytes 22 and 23 was defined in a previous standard.

The SERVICE ACTION RESERVATION KEY field contains information needed for ~~three~~ 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 actions, 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 reservation, the SCOPE-SPECIFIC ADDRESS field shall contain the Element address, zero filled in the most significant bytes to fit the field. If the service action is REGISTER, REGISTER and IGNORE EXISTING KEY, or CLEAR or if the scope is a Logical Unit reservation, the SCOPE-SPECIFIC ADDRESS field shall be set to zero.

The Activate Persist Through Power Loss (APTPL) bit shall be valid only for ~~the~~ a REGISTER service action. 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 service action, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense data 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 5.3.2.3). 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 5.3.2.1).

Table ~~46~~ 50 summarizes which fields are set by the application client and interpreted by the device server for each service action and scope value. ~~Two~~ The APTPL PERSISTENT RESERVE OUT parameter~~s~~ is ~~are~~ not summarized in table ~~46~~ 50, ~~; the RESERVATION KEY field and the APTPL bit,~~ since it ~~they are~~ is specified above.

Table 50 — PERSISTENT RESERVE OUT Service actions and valid parameters

Service action	**Allowed SCOPE**	**Parameters**			
		TYPE	**RESERVATION KEY**	**SERVICE ACTION RESERVATION KEY**	**SCOPE-SPECIFIC ADDRESS**
REGISTER	ignored	ignored	valid	valid	ignored
REGISTER and IGNORE EXISTING KEY	ignored	ignored	ignored	valid	ignored
RESERVE	Logical Unit Element	valid valid	valid	ignored ignored	ignored valid (Element)
RELEASE	Logical Unit Element	valid valid	valid	ignored ignored	ignored valid (Element)
CLEAR	Ignored	ignored	valid	ignored	ignored
PREEMPT	Logical Unit Element	valid valid	valid	valid valid	ignored valid (Element)
PREEMPT & CLEAR	Logical Unit Element	valid valid	valid	valid valid	ignored valid (Element)

7.13 PREVENT ALLOW MEDIUM REMOVAL command

For an initiator that has executed a PERSISTENT RESERVE OUT command with a service action of RESERVE, ~~or~~ REGISTER, or REGISTER and IGNORE EXISTING KEY, the PREVENT field shall be set to zero as part of the uninterrupted sequence of events performed by a PERSISTENT RESERVE OUT command with a service action of PREEMPT AND ABORT using that initiator's registration value in the SERVICE ACTION RESERVATION KEY field. This allows an initiator to override the prevention of medium removal function for an initiator that is no longer operating correctly.