Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900

May 29, 1998 T10/98-124, revision 1

To: NCITS Technical Committee T10

From: Bob Snively

Subject: SPC-2, Persistent Reservation: Additional proposed corrections

During implementation of various persistent reservation functions, the SPC standard has been found to be incomplete or unclear in a number of places. SPC-2 is the ideal place to correct these problems. In addition, the architectural study required to implement persistent reservation has revealed that some additional functions are required. This document proposes the necessary changes to SPC-2 and additionally proposes changes to SBC. My thanks to the many people who have contributed to this study, including Gerry Houlder, Andy Hisgen, and Brian Edgar.

Revision 1 contains those modifications requested by the NCITS T10 working group on March 18, 1998. Most of these changes are minor adjustments in wording to more fully express the intent. All changes in this document are marked with change bars. I have removed comments on the treatment of reservations by SPC and SBC commands and prepared a separate document to include that information.

1 Unregister capability

The service action description for the Register service action (7.13.1.1) needs to be clarified to clearly indicate that the unregister function is performed by performing a Register service action with the Service Action Reservation Key set to a zero value. At present, it is difficult to determine whether the reservation registration key resources are released by that action or not. This requires a change to 7.3.1.1 as shown below. Similar wording is required in other sections which are also indicated in this document.

7.13.1.1 Register

The PERSISTENT RESERVE OUT command executing a Register service action registers a reservation key with a device server without generating a reservation. A reservation key of zero has a special meaning as defined in item e below. For each initiator that performs a PERSISTENT RESERVE OUT 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 Register service action from the same initiator ~~or until the key is reset to the default value of zero~~ or until the initiator registration is removed by one of the following actions:

a) powering down the logical unit, if the last APTPL received by the device server was zero (see 7.13.2);

b) performing a Clear service action;

c) performing a Preempt service action~~; or~~

d) performing a Preempt and Clear service action; or

e) performing a Register service action with the value of the Service Action Reservation key set to zero.

When a reservation key has not yet been established or when the reservation key has been removed, ~~the default~~ a reservation key of zero shall be used when the initiator performs a PERSISTENT RESERVE OUT with the register service action. ~~registers a reservation key again~~. When the reservation key has been removed, no information is reported for the initiator in the Report Keys service action and all persistent reservations associated with that initiator and reservation key shall be released.

2 Clarify CDB field requirements

The scope field and the type field are not defined and have no meaning for the Register and for the Clear service actions. These fields shall be ignored by the device server.

In the last paragraph of section 7.13.1.1, describing the Register service action, the following change should be made.

The Register service action may be performed regardless of any active persistent reservations. All existing persistent reservations for the initiator receive the new reservation key. The Scope field and the Type field shall be ignored by the device server.

After the third paragraph of section 7.13.1.4, describing the Clear service action, the following change should be made.

The Scope field and the Type field for a Clear service action shall be ignored by the device server.

Table 47 needs to be updated to make interpreting these paragraphs simpler. A suggested format for the changes is provided below:

Table 47 PERSISTENT RESERVE OUT Service actions and valid parameters

Service action	Parameters			
	Scope allowed	Type	Service Action Reservation key	Element or Element Parameters
Register	ignored	ignored	valid	ignored
Reserve Reserve Reserve	LU Extent Element	valid valid valid	ignored ignored ignored	ignored Extent valid Element valid
Release Release Release	LU Extent Element	valid valid valid	ignored ignored ignored	ignored Extent valid Element valid
Clear	ignored	ignored	ignored	ignored
Preempt Preempt Preempt	LU Extent Element	valid valid valid	valid valid valid	ignored Extent valid Element valid
Preempt & clear Preempt & clear Preempt & clear	LU Extent Element	valid valid valid	valid valid valid	ignored Extent valid Element valid

3 Proposed improvements to Release service action

The following text is proposed to clarify what actually is performed when a Release service action is performed when no such reservation exists. The text is modified in section 7.13.1.3, paragraph 1. In the first paragraph, the text is modified to clarify that there is no release performed when there is no reservation to be released. In the second paragraph, the modifications clarify the requirements for checking that a release service action is performed against a properly matching reservation. Item a) in the third paragraph is modified to indicate that power off actually removes reservation keys and releases any resources required to preserve them.

The PERSISTENT RESERVE OUT command performing a Release service action removes an active persistent reservation held by the same initiator. The fields associated with the Release service action shall match fields of the active persistent reservation. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Release service action when no persistent res-

ervation exists from that initiator. In this case, the device server shall return GOOD status without altering any other reservation. The reservation key shall not be changed by the Release service action.

The device server shall return a CHECK CONDITION status for a PERSISTENT RESERVE OUT command that specifies the release of a persistent reservation matching the requesting initiator's ID and some but not all of the scope, reservation key, reservation type, and extent values. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID RELEASE OF ACTIVE PERSISTENT RESERVATION. Attempts to release persistent reservations where none of the scope, reservation key, reservation type, and extent values match an existing persistent reservation held by the requesting initiator shall not be considered errors.

An active persistent reservation may also be released by either of the following mechanisms:

a) Power off. When the most recent APTPL value received by the device server is zero (see 7.13.2), a power off performs a hard reset, clears all persistent reservations, and removes all registered reservation keys~~, setting them to the default value of zero~~ (see 7.13.1.1); or

b) Execution of a PERSISTENT RESERVE OUT command from another initiator with a Persistent Reserve service action of Preempt or Preempt and Clear.

The remaining paragraphs of 7.13.1.3 are unchanged.

4 Proposed clarification of Clear service action

The text of 7.13.1.4 needs to be clarified to indicate that the removal of a reservation key does not simply reset the value, but actually removes the reservation key and any resources related to the key.

The PERSISTENT RESERVE OUT command that successfully performs a Clear service action shall remove all persistent reservations for all initiators. All reservation keys shall be removed~~ and reset to the default value of zero~~ (see 7.13.1.1). Any commands from any initiator that have been accepted by the device server as nonconflicting shall continue normal execution.

5 Clarification of details in the Preempt service action

Paragraph 1 of section 7.13.1.5 needs to explicitly indicate that the preempting of a persistent reservation creates a reservation by the initiator performing the preempt service action.

The PERSISTENT RESERVE OUT command that successfully performs a Preempt service action shall remove all persistent reservations for all initiators that are registered with the Service Action Reservation key specified in

the PERSISTENT RESERVE OUT parameter list. It shall also establish a persistent reservation for the preempting initiator. Any commands from any initiator that have been accepted by the device server as nonconflicting shall continue normal execution. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Preempt service action when no persistent reservation exists for the initiator identified by the Service Action Reservation key.

The fourth paragraph of section 7.13.1.5 needs to be clarified to show that the removal of a reservation key does not simply reset the value, but actually removes the reservation key and any resources related to the key. In addition, this paragraph should indicate that execution of the Preempt service action with a Service Action Reservation Key equal to its own reservation key shall establish the new specified reservation. Other initiators having the same reservation key shall have the their reservation key removed.

The registration key for ~~the~~ other initiators that have been preempted shall be removed ~~and reset to the default value of zero~~ (see 7.13.1.1) by the Preempt service action. The reservation key for an initiator that has performed a Preempt service action with its own reservation key specified in the Service Action Reservation Key shall remain unchanged, although all other specified releasing actions and reservation actions shall be performed.

6 Clarification of details in the Preempt and Clear service action

Paragraph 1 of section 7.13.1.6 needs to explicitly indicate that the preempting of a persistent reservation creates a reservation by the initiator performing the Preempt and Clear service action.

The PERSISTENT RESERVE OUT command performing a Preempt and Clear service action removes all persistent reservations for all initiators that are registered with the Service Action Reservation key specified by the PERSISTENT RESERVE OUT parameter list. It also establishes a persistent reservation for the preempting initiator. Every command from the initiators being preempted shall be terminated as if an ABORT TASK task management function had been performed by the preempted initiator. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Preempt and Clear service action when no persistent reservation exists for the initiator identified by the Service Action Reservation key. However, if the key is registered, the Clear portion of the action shall execute normally.

The sixth paragraph of section 7.13.1.5 needs to be clarified to show that the removal of a reservation key does not simply reset the value, but actually removes the reservation key and any resources related to the key. In addition, this paragraph should indicate that execution of the Preempt and Clear service action with a Service Action Reservation Key equal to its own reservation key shall clear any commands for that device and establish the new

specified reservation. Other initiators having the same reservation key shall have the their reservation key removed.

The reservation key registered for ~~the~~ other initiators that have been preempted shall be removed ~~and reset to the default value of zero~~ (see 7.13.1.1) by the Preempt and Clear service action. The reservation key for an initiator that has performed a Preempt and Clear service action with its own reservation key specified in the Service Action Reservation Key shall remain unchanged, although all other specified clearing actions, releasing actions, and reservation actions shall be performed.

7 Correction of reservation key removal text in APTPL description

The sixth paragraph of section 7.13.2, needs to indicate that, if APTPL is zero, the reservation keys and related resources are actually removed by a power off, not merely set to the default value.

If the last valid APTPL bit value received by the device server is zero, the loss of power in the target shall release all persistent reservations and remove all reservation keys~~, setting them to the default value of zero~~ (see 7.13.1.1). If the last valid APTPL bit value received by the device server is one, the logical unit shall retain all persistent reservations and all reservation keys for all initiators even if power is lost and later returned. The most recently received valid APTPL value from any initiator shall govern logical unit's behavior in the event of power loss.

8 Exhaustion of reservation resources

At present, the SPC-2 document does not specify what action shall be taken if an attempt is made to create a reservation, but all reservation resources (memory space, allocated disk space, or other resources) are already used. The following addition after the last paragraph of section 7.13 is proposed.

If a PERSISTENT RESERVE OUT command 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 ASC/ASCQ shall be set to INSUFFICIENT RESERVATION RESOURCES (55/02).

9 Clarify actual exceptions to RESERVATION CONFLICT

The text of 7.13 is a little too restrictive in indicating what operations may proceed without encountering a RESERVATION CONFLICT. In fact, there are reservations that may cause RESERVATION CONFLICT even with Preempt or Preempt and Clear service actions. As an example, a Preempt operation that creates a new but conflicting reservation for the target performing the command will fail with RESERVATION CONFLICT status. In addition,

certain commands, including INQUIRY and REQUEST SENSE, will not conflict with any reservation. The following suggested modifications clarify this.

The second paragraph of section 7.12 is modified to show how reservations conflict with the PERSISTENT RESERVE IN command.

When a device server receives a PERSISTENT RESERVE IN command and RESERVE(10) or RESERVE(6) logical unit or extent reservations or SMC element reservations are active (see 7.21), the command shall be rejected with a RESERVATION CONFLICT status. A PERSISTENT RESERVE IN command shall not conflict with any persistent reservation.

The second paragraph of section 7.13 is modified and split into several sections to show how reservations conflict with the PERSISTENT RESERVE OUT command.

The PERSISTENT RESERVE OUT command ~~Persistent reservations~~ shall conflict with any reservations established by the RESERVE command.

A PERSISTENT RESERVE OUT command with a service action of Preempt shall not conflict with any persistent reservation for the initiator or initiators being preempted. If there is a reservation by any initiator except the initiator or initiators being preempted that conflicts with the new reservation being formed by the initiator executing the PERSISTENT RESERVE OUT command, the command shall end with RESERVATION CONFLICT status and the Preempt service action shall not be performed.

A PERSISTENT RESERVE OUT command with a service action of Preempt and Clear shall not conflict with any persistent reservation for the initiator or initiators being preempted and cleared. If there is a reservation by any initiator except the initiator or initiators being preempted and cleared that conflicts with the new reservation being formed by the initiator executing the PERSISTENT RESERVE OUT command, the command shall end with RESERVATION CONFLICT status and the Preempt and Clear service action shall not be performed.

A PERSISTENT RESERVE OUT command with a service action of Reserve shall end with RESERVATION CONFLICT status and the reservation shall not be made if the new reservation conflicts in scope, type, or extent with any persistent reservation by any initiator at the time the PERSISTENT RESERVE OUT is enabled for execution.

A PERSISTENT RESERVE OUT command with a service action of Release or a service action of Clear shall not conflict with any reservation by any initiator.

A PERSISTENT RESERVE OUT command with any service action except Register shall end with RESERVATION CONFLICT status if the reserving initiator does not have a registered reservation key.

Initiators performing PERSISTENT RESERVE OUT Service actions are identified by a reservation key provided by the application client. An application client may use the PERSISTENT RESERVE IN command to identify which initiators are holding conflicting or invalid persistent reservations and use the PERSISTENT RESERVE OUT command to preempt those reservations if required.

The fifth paragraph of section 7.13 can now be deleted as follows:

~~Commands from any initiator that conflict with a successfully established persistent reservation shall be rejected with a status of RESERVATION CONFLICT with the following exceptions:~~

~~a) PERSISTENT RESERVE IN shall not conflict with any persistent reservation;~~

~~b) PERSISTENT RESERVE OUT with a service action of Preempt shall not conflict with any persistent reservation; and~~

~~c) PERSISTENT RESERVE OUTwith a service action of Preempt and Clear shall not conflict with any persistent reservation.~~

Note that I have resolved one possible ambiguity somewhat arbitrarily. I have allowed any conflicting reservation for initiators other than an initiator being preempted or preempted and cleared to block the execution of the PERSISTENT RESERVE OUT command with a reservation conflict. Systems must select the reservations to be performed very carefully so that the preempt operations used to recover control are not blocked when multiple reservations are outstanding and more than one initiator must be preempted. I would recommend that only non-exclusive types of reservations be used if multiple hosts may hold a reservation at the same time. Extent reservations are especially interesting, since many can be held from many initiators. A LUN reservation may conflict with one or more of those reservations, creating a deadlock. Registrants only types of reservations are especially forgiving, since only one reserving initiator is required.

Because the text of the PERSISTENT RESERVE OUT command now completely defines the reservation conflicts, the corresponding text should be removed from the individual service action subsections as follows:

7.13.1.2, second paragraph removed:

~~A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Reserve service action that conflicts with any active persistent reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT~~

~~is enabled for execution. The PERSISTENT RESERVE OUT command with a Reserve service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.~~

7.3.1.3, next to last paragraph removed:

~~The PERSISTENT RESERVE OUT command with a Release service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.~~

7.3.1.4, third paragraph removed:

~~The PERSISTENT RESERVE OUT command with a Clear service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.~~

7.13.1.5, fifth paragraph removed:

~~A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Preempt service action that conflicts with any active persistent reservations except the preempted reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT is enabled for execution. The PERSISTENT RESERVE OUT command with a Preempt service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.~~

7.13.1.6, , eighth paragraph removed:

~~A status of RESERVATION CONFLICT shall be generated for a PERSISTENT RESERVE OUT command that specifies the execution of a Preempt and Clear service action that conflicts with any active persistent reservations except the preempted reservations from the same initiator in scope, type, or extent at the time the PERSISTENT RESERVE OUT is enabled for execution. The PERSISTENT RESERVE OUT command with a Preempt service action shall be rejected with a status of RESERVATION CONFLICT if the initiator requesting the command has not previously performed a Register service action with the device server.~~

10 Clarification of requirements on parameters

The paragraph just above table 47 in section 7.13.2 should be clarified to indicate that those parameters that are ignored shall not be checked by the device server. The ignored parameters may have any value. After

consideration, the committee felt that it was simplest to include this as one of the keywords in section 3.3. In that case, the paragraph just above table 47 would be rewritten as follows:

Table 47 summarizes which fields are set by the application client and interpreted by the device server for each Service action and Scope value. Two PERSISTENT RESERVE OUT parameters are not summarized in table 47; Reservation key and APTPL, since they are specified above.

The word “ignored” should be included in the keyword portion of SPC-2, for use by table 47 and other similar paragraphs. The following text is proposed paragraph after section 3.3.2:

3.3.x ignored: A parameter that is specified as ignored shall not be examined by the SCSI device receiving the parameter and may be set to any value by the SCSI device transmitting the value.

11 Clarification of persistent reservation management method

The capability of devices using persistent reservation to create various kinds of shared reservations conditioned by the registry of a reservation key is a significant extension of the reservation model and should be included in section 5.3.2. The following text is proposed:

The Persistent Reservations management method is used among multiple initiators that require operations to be protected across initiator failures, which usually involve hard resets. Persistent reservations persist across recovery actions, to provide initiators with more detailed control over reservations recovery. Persistent reservations for failing initiators may be preempted by another initiator as part of the recovery process. Persistent reservations are retained by the device server until released, preempted, or until cleared by mechanisms specified in this standard. Persistent reservations are optionally retained when power to the target is lost.

In addition to the reservations created directly by an initiator, the Persistent Reservations management method allows the creation of reservations which become automatically shared among those initiators which have registered with the device server.

12 Atomic operation of reservations

It is important that each service action performed by a PERSISTENT RESERVE OUT command be performed as a single operation without being modified by or conflicting with other PERSISTENT RESERVE OUT commands that may also be requested. While this is made evident in several parts of sections 7.12 and 7.13, there are at least a few places where it should be made more explicit.

13 Clarification of reservation conflicts for SPC-2 and SBC commands.

These sections created significant discussion and will probably involve re-writing a number of reservation defining paragraphs throughout SPC-2. The text of revison 1 of this document is available for reference, but in the interests of moving this proposal forward, I am leaving the text out of this proposal. A separate proposal will be created to correct the many small discrepancies in this area. An erratum may be required for SBC.