

Accredited Standards Committee
NCITS, National Committee for Information Technology Standards

Doc: T10-218r0
Date: July 16, 1997
Project: FC-AL-2
Ref Doc.:
Reply to: Bob Snively

To: T10 Membership
From: Bob Snively

Subject: Review of Persistent Reservation

During review of the persistent reservation, three issues were identified which should be resolved. Those issues and proposed solutions are discussed in this document.

Conditions for performing persistent reservation operation

Overview:

There is a phrase in the discussion about creating reservation conflicts with the PERSISTENT RESERVE OUT command that has profound implications.

Commands from any initiator that conflict with a successfully established persistent reservation shall be rejected with a status of RESERVATION CONFLICT. The following commands shall not conflict with a reservation established by the PERSISTENT RESERVE OUT command:

PERSISTENT RESERVE IN

PERSISTENT RESERVE OUT (with an Service action of Preempt)

PERSISTENT RESERVE OUT (with an Service action of Preempt and Clear)

PERSISTENT RESERVE OUT (with a Reserve service action that does not conflict with established persistent reservations or tasks)

The offending words here are "or tasks". That implies what is not stated anywhere else that I can find, that a PERSISTENT RESERVE OUT can only be executed without a RESERVATION CONFLICT after all queued tasks that may conflict with the reservation are completed. The words apparently apply to all tasks, including those that are enabled by having been started and those that are blocked or queued even against an ordered PERSISTENT RESERVE OUT. Not only is the PERSISTENT RESERVE OUT potentially delayed by a task, but it can even encounter a RESERVATION CONFLICT with the task.

Note 26 (below) seems to imply the behavior I would actually expect. That is, when a PERSISTENT RESERVE OUT conflicts with a task that has already been passed to the device or is passed to the device before the PERSISTENT RESERVE OUT command is enabled and begins execution, the occurrence of a RESERVATION CONFLICT status and the time of effectiveness of a reserve service action is vendor specific.

Notes 27 and 29 provide similar warnings for Preempt and Preempt & Clear.

NOTE 26 For the simplest predictable behavior, the Reserve service action should be performed with the Ordered task attribute.

Suggested Change:

At the earliest time allowed by the standards process:

I would suggest that we remove the words "or tasks" from the document.

I would suggest that we should make NOTES 26, 27, and 29 a little less cryptic and state:

NOTE xx: The time of effectiveness of a reservation with respect to other tasks being managed by the device server is vendor specific. Successful completion of the PERSISTENT RESERVE OUT command indicates that the new reservation is active. An active reservation may apply to some or all tasks queued before the completion of the PERSISTENT RESERVE OUT command. The reservation shall apply to all tasks received after successful completion of the PERSISTENT RESERVE OUT command.

Duplicate reservations

Overview:

Non-duplicate reservations are simple for a target to manage. It is easy enforce the rule that each must be cleared separately. The text of 7.13.1.2 makes this very clear:

New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The persistent reservation of a logical unit or the persistent reservation of extents having the same type value shall be permitted if no conflicting persistent reservations are held by another initiator. When such overlapping persistent reservations are released, each of the extent reservations and the logical unit reservation shall be removed with a separate Release service action.

However, two identical reservations from the host have a less clear behavior. In particular, when a release comes through, should the first, the second, or both be cleared? I expect that in many implementations, there may not even be a simple mechanism to count and compare the number of executions of identical Reserve service actions and Release service actions.

Suggested Change:

At the earliest opportunity in the standardization process, I suggest we make the following changes.

I would propose that the paragraph be expanded to allow a Release service action to release all identical reservations. With that addition, it will not matter if individual copies of the reservations are kept or not. It will also not matter how many Release service actions occurred relative to the number of Reserve service actions. Text is already present to indicate that the release of reservations that do not exist is not an error. The new text would read:

New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The persistent reservation of a logical unit or the persistent reservation of extents having the same type value shall be permitted if no conflicting persistent reservations are held by another initiator. When such overlapping persistent reservations are released, each of the extent reservations and the logical unit reservation shall be removed with a separate Release service action. Multiple identical reservations from the same initiator are all simultaneously released by a single Release service action that matches the reservations.

Note that there is one peculiar result of this. If you have independent control processes operating against the same LU from the same initiator, then one of those control processes may be very much surprised to find that its reservations have been removed by another control process. That is probably bad program design, rather than a SCSI architecture error.

Reservation key clearing

Overview:

At present, text in 7.13.1.1 says:

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 by powering down the logical unit, if the last APTPL received by the device server was zero (see 7.13.2) or by performing a Clear, Preempt, or Preempt and Clear service action.

Similar text is found in
7.13.1.3, Release, item a)

- 7.13.1.4, Clear, first paragraph
- 7.13.1.5, Preempt, 4th paragraph
- 7.13.1.6, Preempt and Clear, 7th paragraph
- 7.13.2, PERSISTENT RESERVE OUT parameter list, 6th paragraph

It is never explicitly indicated what "reset to the default value of zero" really means. I believe that it was intended to mean that the reservation key is unregistered and disassociated from any relationship to an initiator port. At the same time, any new registration to the device from the unregistered initiator would use the default reservation key of zero. That is a necessary behavior to properly handle the proposed "Write Exclusive, Registrants Only" reservation type. If the state of being registered is not cleared by the action of resetting the reservation key, then an offending initiator cannot be locked out. In addition, extinct initiator tables cannot be recovered, but must remain available for setting to a non-default value.

Suggested Change:

I suggest that the following changes be installed in the document at the earliest time allowed by the standardization process.

I suggest that the text be a little more specific in its definition of resetting the reservation, explicitly indicating that the initiator becomes unregistered and any resources associated with the initiator be released. The demonstration paragraph above (7.13.1.1) would be changed to read:

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 initiator registration is removed by one of the following actions:

- 1) powering down the logical unit, if the last APTPL received by the device server was zero (see 7.13.2)
- 2) performing a Clear service action
- 3) performing a Preempt service action
- 4) performing a Preempt and Clear service action

When the reservation has been removed, the default reservation key of zero shall be used when the initiator registers a reservation key again. When the reservation has been removed, no information is reported for the initiator in the Report Keys service action.

Similar text indicating that "the initiator registration is removed" would be installed in the following locations:

- 7.13.1.3, Release, item a)
- 7.13.1.4, Clear, first paragraph
- 7.13.1.5, Preempt, 4th paragraph
- 7.13.1.6, Preempt and Clear, 7th paragraph
- 7.13.2, PERSISTENT RESERVE OUT parameter list, 6th paragraph

Sincerely,

Robert Snively
Phone: (415) 786-6694
Email: bob.snively@sun.com