

Date: May 07, 2003
To: T10 Committee (SCSI)
From: George Penokie (IBM/Tivoli)
Subject: Yet More Persistent Reservation Fixes

1 Overview

During an internal review of persistent reservation several errors and inconsistencies were discovered. This proposal points out those areas and suggests corrections to SPC-3 rev 12.

This revision has changed the meaning of a zero value for the service action reservation key from not removing any registrations to removing all registrations for an all registrants preempt.

2 Corrections

2.1 General

The term << initiator >> and << initiator port >> in most, if not all, cases should be changed to << I_T nexus >> in the reservation model and reservation commands. Basically, initiator ports don't register or get registered; application clients register I_T nexuses.

Note: This comment is from Rob Elliott but I agree with it.

2.2 Table 32 and Table 33

In Table 32 and Table 33 service actions that are allowed in the presence of various reservations the column titled << Addressed LU has a persistent reservation held by another initiator >> should be changed to << Addressed LU has a persistent reservation and the SCSI initiator port is not a reservation holder >>. This change is required because to the all registrants option.

Also in table 32 and table 33 all the references to << initiator >> should be changed to << SCSI initiator port >>.

2.3 5.5.2.1 Overview of the Persistent Reservations management method

In the paragraph:

Reservation key values may be used by application clients to identify initiator ports, using application specific methods that are outside the scope of this standard. This standard provides the ability to register no more than one key per I_T_L nexus. **Multiple initiator ports may use the same key for a logical unit accessed through the same target port.** An initiator port may establish registrations for multiple logical units in a SCSI target device using any combination of unique or duplicate keys. These rules provide the ability for an application client to preempt multiple initiator ports with a single PERSISTENT RESERVE OUT command, but they do not provide the ability for the application client to uniquely identify the initiator ports using the PERSISTENT RESERVE commands.

The statement:

<< Multiple initiator ports may use the same key for a logical unit accessed through the same target port. >> says nothing about multiple port targets. It should be changed to << Multiple initiator ports may use the same key value for a logical unit accessed through the same target ports. An initiator port may use the

same key value for a logical unit accessed through different target ports. A separate key shall be maintained for each I_T nexus, regardless of the key's values. >>

2.4 5.5.2.2 Preserving persistent reservations and registrations

In the paragraph:

When nonvolatile memory has not become ready since **a power cycle, other than those listed above shall return** CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense code shall be set as described in table 166 (see 6.29).

This statement:

<< a power cycle, other than those listed above shall return >> should be << a power cycle, commands other than those listed above shall return >>

2.5 5.5.2.3.2 Reporting reservation keys

In this section and others the term << initiator port >> needs to be looked at to see if it should be changed to I_T nexus. For example in the paragraph:

An application client may issue a PERSISTENT RESERVE IN command with READ KEYS service action to determine if any **initiator ports** have registered with a logical unit.

In this case initiator port is really an I_T nexus.

As a result the statement:

<< An application client may issue a PERSISTENT RESERVE IN command with READ KEYS service action to determine if any **initiator ports** have registered with a logical unit. >>

Should be changed to:

An application client may issue a PERSISTENT RESERVE IN command with READ KEYS service action to determine if any I_T nexus have been registered with a logical unit through any target port.

Also it is not clear all keys should be reported regardless of the target port through which they were set up.

The statement:

b) The reservation key for every initiator port that is currently registered.

Should be changed to:

b) The reservation key for every I_T nexus that is currently registered regardless of the SCSI target port through which the registration occurred.

2.6 5.5.2.6 The persistent reservation holder

The title of this section should be changed to << Persistent reservation holder >>.

2.7 5.5.2.7.1.1 Handling for released registrants only persistent reservations

In the paragraph:

If the TYPE or SCOPE changed or the reservation was released, the device server shall establish a unit attention for each registered initiator port whose reservation key was not removed. The additional sense code shall be set as follows:

Need to add in the words about << except for the SCSI initiator port that issued the command >> to the end of the first sentence so it reads << If the TYPE or SCOPE changed or the reservation was released, the device server shall establish a unit attention for each registered SCSI initiator port whose reservation key was not removed except for the SCSI initiator port through which the command was issued. >>

2.8 5.5.2.7.1.2 Handling for released all registrants persistent reservations

In the paragraph:

If a persistent reservation was removed or changed, the device server shall establish a unit attention for each registered initiator port whose reservation key was not removed. The additional sense code shall be set as follows:

Need to add in the words about << except for the SCSI initiator port that issued the command >> to the end of the first sentence so it reads << If a persistent reservation was removed or changed, the device server shall establish a unit attention for each registered SCSI initiator port whose reservation key was not removed except for the SCSI initiator port through which the command was issued. >>

2.9 5.5.2.7.4.1 Overview of preempting

Figure 3 — Device server interpretation of PREEMPT service action

Add the statement << Remove all registrations >> to the flow chart in as shown below.

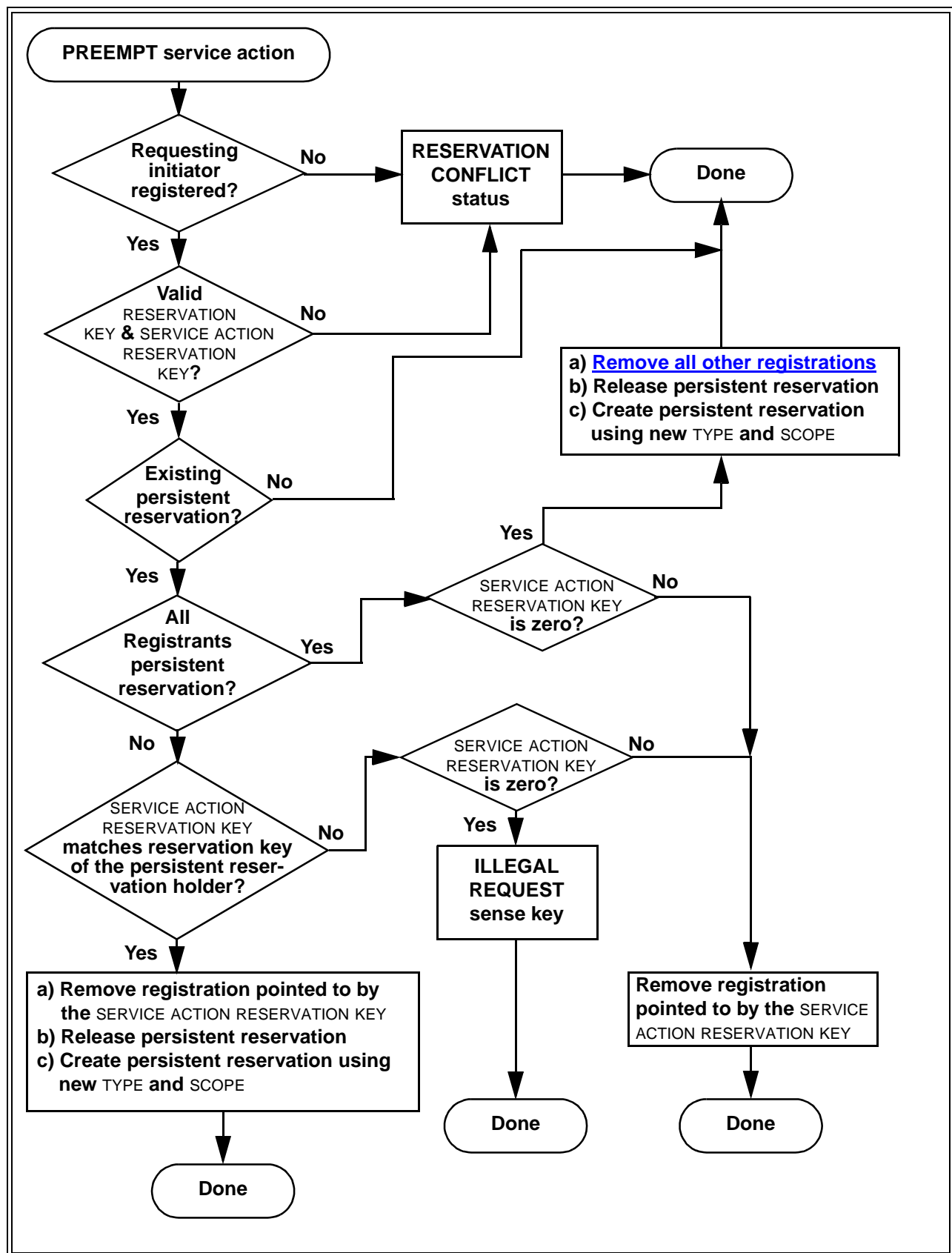


Figure 1 — Device server interpretation of PREEMPT service action

2.10 5.5.2.7.4.3 Preempting persistent reservations and removing registrations

The title of this section should be changed to:

Preempting persistent reservations and registration handling

The paragraph:

If the SERVICE ACTION RESERVATION KEY identifies the a persistent reservation holder (see 5.5.2.6), the device server shall perform a preempt by doing the following as an uninterrupted series of actions:

The statement << identifies the a persistent >> should be << identifies a persistent >>

There also needs to be a clearer statement about what happens when an All Registrants is in place and a preempt is received with a zero key.

The statement:

b) Remove the registrations for all initiator port or initiator ports identified by the SERVICE ACTION RESERVATION KEY, except the initiator port that issued the PERSISTENT RESERVE OUT command;

Should be changed to:

b) Remove the registrations for all initiator port or initiator ports identified by the SERVICE ACTION RESERVATION KEY, except the initiator port that issued the PERSISTENT RESERVE OUT command.<< If an all registrants persistent reservation is present and the SERVICE ACTION RESERVATION KEY is set to zero then all registrations shall be removed except for the initiator port that issued the PERSISTENT RESERVE OUT command; >>

2.11 5.5.2.7.4.4 Removing registrations

In the paragraph:

If the SERVICE ACTION RESERVATION KEY field does not identify a persistent reservation holder the device server shall perform a preempt by doing the following in an uninterrupted series of actions:

Add the following words to make it clear that a registration is only removed if there is a persistent reservation in place as shown in figure 3:

<< If there is a persistent reservation and if the SERVICE ACTION RESERVATION KEY field does >>.

In addition a paragraph needs to be added to this section that states << If there is no persistent reservation there shall be no change to any registrations and a GOOD status shall be returned. >>

There also needs to be a paragraph added to this section that states:

<< It is not an error for a PERSISTENT RESERVE OUT with a PREEMPT service action or a PREEMPT AND ABORT service action to set the RESERVATION KEY and the SERVICE ACTION RESERVATION KEY to the same value, however, no unit attention condition is established for the SCSI initiator port that sent the PERSISTENT RESERVE OUT command. >>

2.12 5.5.2.7.5 Preempting and aborting

The paragraph:

The actions described in the preceding list shall be performed for all initiator ports that are registered with the service action reservation key value, without regard for whether the preempted initiator port(s) hold the persistent reservation.

Should be changed to the following to all for the all registrants case of a zero value in SERVICE ACTION RESERVATION KEY field.

The actions described in the preceding list shall be performed for all I_T nexus that are registered with the service action reservation key value, without regard for whether the preempted initiator port(s) hold the persistent reservation. If an all registrants persistent reservation is present and the service action reservation key value is set to zero the device server shall abort all tasks for all registered I_T nexus.

2.13 7.12.3 PERSISTENT RESERVE OUT parameter list

In the paragraph:

If the SPEC_I_PT (Specify Initiator Ports) bit is set to zero, the device server shall ignore the additional parameter data and shall apply the registration only to the initiator port that sent the PERSISTENT RESERVE OUT command. If the SPEC_I_PT bit is set to one for the REGISTER or REGISTER AND IGNORE EXISTING KEY service actions, the additional parameter data shall include a list of transport IDs (see table 89) and the device server shall apply the registration to each initiator port specified by a TransportID. **If a registration fails for any initiator port**, none of the other registrations shall be made.

The statement << If a registration fails for any initiator port, >> should indicate an instance that could result in a failure. I suggest changing it to << If a registration fails for any SCSI initiator port (e.g., target device does not have enough resources available to hold the registration information), >>