Date: April 19, 2007

To: T10 Committee (SCSI)

From: George Penokie (IBM)

Subject: SPC-4: Duplicate persistent reservation wording removal

1 Overview

This proposal addresses the following note from proposal 07-080:

Editor's Note 1: Additional issue that might be addressed in a follow-on proposal or another revision of this proposal: All the rules in 5.6.10.1.2, 5.6.10.1.3, and 5.6.10.1.4(all under the Overview subclause) seem to be duplicates of rules in 5.6.10.2 (RELEASE), 5.6.10.3 (REGISTER), 5.6.10.4 (PREEMPT), 5.6.10.5 (PREEMPT AND ABORT), and 5.6.10.6 (CLEAR). The 5.6.10.x sections might be best removed entirely.

Note to the Editor: The text indicated by mangenta contain no changes from the existing text and is only highlighted for reference.

2 SPC-4 changes

5.6.10.1 Overview

5.6.10.1.1 Summary of service actions that release persistent reservations and remove registrations

An application client may release or preempt the persistent reservation by issuing one of the following commands through a registered I_T nexus with the RESERVATION KEY field set to the reservation key value that is registered with the logical unit for that I_T nexus:

a) A PERSISTENT RESERVE OUT command with RELEASE service action from a persistent reservation holder (see 5.6.10.2);
b) A PERSISTENT RESERVE OUT command with PREEMPT service action specifying the reservation key of the persistent reservation holder or holders (see 5.6.10.4);
c) A PERSISTENT RESERVE OUT command with PREEMPT AND ABORT service action specifying the reservation key of the persistent reservation holder or holders (see 5.6.10.5);
d) A PERSISTENT RESERVE OUT command with CLEAR service action (see 5.6.10.6); or
e) If the I_T nexus is the persistent reservation holder and the persistent reservation is not an all registrants type, then a PERSISTENT RESERVE OUT command with REGISTER service action or REGISTER AND IGNORE EXISTING KEY service action with the SERVICE ACTION RESERVATION KEY field set to zero (see 5.6.10.3).

An application client may remove registrations by issuing one of the following commands through a registered I_T nexus with the RESERVATION KEY field set to the reservation key value that is registered with the logical unit for that I_T nexus:

a) A PERSISTENT RESERVE OUT command with PREEMPT service action with the SERVICE ACTION RESERVATION KEY field set to the reservation key (see 5.6.10.4) to be removed;
b) A PERSISTENT RESERVE OUT command with PREEMPT AND ABORT service action with the SERVICE ACTION RESERVATION KEY field set to the reservation key (see 5.6.10.5) to be removed;
c) A PERSISTENT RESERVE OUT command with CLEAR service action (see 5.6.10.6); or
d) A PERSISTENT RESERVE OUT command with REGISTER service action or REGISTER AND IGNORE EXISTING KEY service action with the SERVICE ACTION RESERVATION KEY field set to zero (see 5.6.10.3).

When a reservation key (i.e., registration) has been removed, no information shall be reported for that unregistered I_T nexus in subsequent READ KEYS service actions until the I_T nexus is registered again (see 5.6.6). As shown in table 1, the processing of any persistent reservation whose persistent reservation holder or holders become unregistered depends on the reservation type.

Table 1 — Processing for released persistent reservations

Reservation Type	~~Reference~~ Processing
Write Exclusive – Registrants Only or Exclusive Access – Registrants Only	~~5.6.10.1.2~~ When the persistent reservation holder (see 5.6.9) of this reservation type becomes unregistered the persistent reservation shall be released.
Write Exclusive – All Registrants or Exclusive Access – All Registrants	~~5.6.10.1.3~~ This persistent reservation shall be released when the registration for the last registered I_T nexus is removed or when the TYPE or SCOPE is changed.
Write Exclusive or Exclusive Access	~~5.6.10.1.4~~ When the persistent reservation holder (see 5.6.9) of this reservation type becomes unregistered the persistent reservation shall be released.

Registrations and persistent reservations may also be released by a loss of power, if the persist through power loss capability is not enabled. When the most recent APTPL value received by the device server is zero (see 6.14.3), a power cycle:

a) Releases all persistent reservations; and
b) Removes all registered reservation keys (see 5.6.6).

~~5.6.10.1.2 Processing for released Registrants Only persistent reservations~~

~~When the persistent reservation holder (see 5.6.9) of a Write Exclusive – Registrants Only or Exclusive Access – Registrants Only type reservation becomes unregistered the persistent reservation shall be released.~~

~~For every I_T nexus whose reservation key is removed, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus and the additional sense code shall be based on the PERSISTENT RESERVE OUT command service action as follows:~~

~~a) If the service action was CLEAR, the additional sense code shall be set to RESERVATIONS PREEMPTED; or~~
~~b) If the service action was PREEMPT or PREEMPT AND ABORT, the additional sense code shall be set to REGISTRATIONS PREEMPTED.~~

~~If the TYPE or SCOPE have changed, then for every I_T nexus whose reservation key was not removed except for the I_T nexus on which the PERSISTENT RESERVE OUT command was received, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus, with the additional sense code set to RESERVATIONS RELEASED. If the TYPE or SCOPE have not changed, then no unit attention condition(s) shall be established for this reason.~~

~~If the reservation was released, then for every I_T nexus whose reservation key was not removed except for the I_T nexus on which the PERSISTENT RESERVE OUT command was received, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus, with the additional sense code set to RESERVATIONS RELEASED. If the reservation was not released, then no unit attention condition(s) shall be established for this reason.~~

~~5.6.10.1.3 Processing for released All Registrants persistent reservations~~

~~A Write Exclusive – All Registrants or Exclusive Access – All Registrants type persistent reservation shall be released when the registration for the last registered I_T nexus is removed or when the TYPE or SCOPE is changed.~~

~~The device server shall establish a unit attention condition for the initiator port associated with every registered I_T nexus whose reservation key was removed, with the additional sense code set as follows:~~

a) ~~If the service action was CLEAR, the additional sense code shall be set to RESERVATIONS PREEMPTED; or~~
b) ~~If the service action was PREEMPT or PREEMPT AND ABORT, the additional sense code shall be set to REGISTRATIONS PREEMPTED.~~

~~If a persistent reservation was released using a RELEASE service action, see 5.6.10.2.~~

~~5.6.10.1.4 Processing for other released persistent reservations~~

~~When the persistent reservation holder (see 5.6.9) of a Write Exclusive or Exclusive Access type reservation becomes unregistered the persistent reservation shall be released.~~

5.6.10.2 Releasing

Only the persistent reservation holder (see 5.6.9) is allowed to release a persistent reservation.

An application client releases the persistent reservation by issuing a PERSISTENT RESERVE OUT command with RELEASE service action through an I_T nexus that is a persistent reservation holder with the following parameters:

a) RESERVATION KEY field set to the value of the reservation key that is registered with the logical unit for the I_T nexus; and
b) TYPE and SCOPE fields set to match the persistent reservation being released.

In response to a persistent reservation release request from the persistent reservation holder the device server shall perform a release by doing the following as an uninterrupted series of actions:

a) Release the persistent reservation;
b) Not remove any registration(s);
c) If the released persistent reservation is a registrants only type or all registrants type persistent reservation, the device server shall establish a unit attention condition for the initiator port associated with every registered I_T nexus other than I_T nexus on which the PERSISTENT RESERVE OUT command with RELEASE service action was received, with the additional sense code set to RESERVATIONS RELEASED; and
d) If the persistent reservation is of any other type, the device server shall not establish a unit attention condition.

The established persistent reservation shall not be altered and the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID RELEASE OF PERSISTENT RESERVATION, for a PERSISTENT RESERVE OUT command that specifies the release of a persistent reservation if:

a) The requesting I_T nexus is a persistent reservation holder (see 5.6.9); and
b) The SCOPE and TYPE fields do not match the scope and type of the established persistent reservation.

If there is no persistent reservation or in response to a persistent reservation release request from a registered I_T nexus that is not a persistent reservation holder (see 5.6.9), the device server shall do the following:

a) Not release the persistent reservation, if any;
b) Not remove any registrations; and
c) Return GOOD status.

5.6.10.3 Unregistering

An application client may remove a registration for an I_T nexus by issuing a PERSISTENT RESERVE OUT command with REGISTER service action or a REGISTER AND IGNORE EXISTING KEY service action with the SERVICE ACTION RESERVATION KEY field set to zero through that I_T nexus.

If the I_T nexus is a reservation holder, the persistent reservation is of an all registrants type, and the I_T nexus is the last remaining registered I_T nexus, then the device server shall also release the persistent reservation.

If the I_T nexus is the reservation holder and the persistent reservation is of a type other than all registrants, the device server shall also release the persistent reservation. If the persistent reservation is a registrants only type, the device server shall establish a unit attention condition for the initiator port associated with every registered I_T nexus, with the additional sense code set to RESERVATIONS RELEASED.

5.6.10.4 Preempting

5.6.10.4.1 Overview

A PERSISTENT RESERVE OUT command with PREEMPT service action or PREEMPT AND ABORT service action is used to:

a) Preempt (i.e., replace) the persistent reservation and remove registrations; or
b) Remove registrations.

Table 2 lists the actions taken based on the current persistent reservation type and the SERVICE ACTION RESERVATION KEY field in the PERSISTENT RESERVE OUT command.

Table 2 — Preempting actions

Reservation Type	Service Action Reservation Key	Action	Reference
All Registrants	Zero	Preempt the persistent reservation and remove registrations.	5.6.10.4.3
	Not Zero	Remove registrations.	5.6.10.4.4
All other types	Zero	Terminate the command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN PARAMETER LIST.	
	Reservation holder's reservation key	Preempt the persistent reservation and remove registrations.	5.6.10.4.3
	Any other, non-zero reservation key	Remove registrations.	5.6.10.4.4

See figure 1 for a description of how a device server interprets a PREEMPT service action to determine its actions (e.g., preempt the persistent reservation, remove registration, or both preempt the persistent reservation and remove registration).

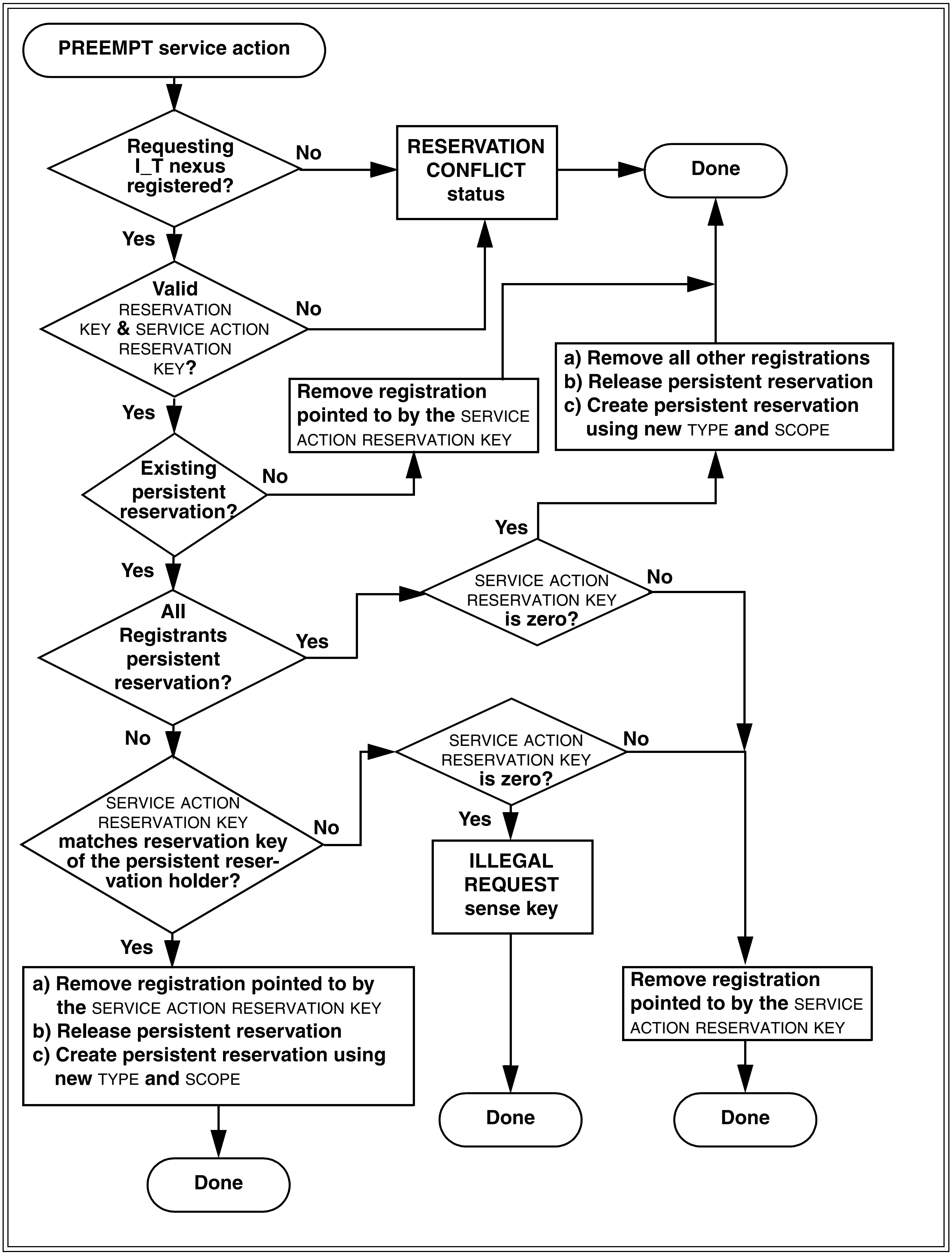
PREEMPT service action
Requesting I_T nexus registered?
No
RESERVATION CONFLICT status
Done
Yes
Valid RESERVATION KEY & SERVICE ACTION RESERVATION KEY?
No
Yes
Remove registration pointed to by the SERVICE ACTION RESERVATION KEY
a) Remove all other registrations
b) Release persistent reservation
c) Create persistent reservation using new TYPE and SCOPE
Existing persistent reservation?
No
Yes
Yes
SERVICE ACTION RESERVATION KEY is zero?
No
All Registrants persistent reservation?
Yes
No
SERVICE ACTION RESERVATION KEY is zero?
No
SERVICE ACTION RESERVATION KEY matches reservation key of the persistent reservation holder?
No
Yes
ILLEGAL REQUEST sense key
Yes
a) Remove registration pointed to by the SERVICE ACTION RESERVATION KEY
b) Release persistent reservation
c) Create persistent reservation using new TYPE and SCOPE
Remove registration pointed to by the SERVICE ACTION RESERVATION KEY
Done
Done
Done

Figure 1 — Device server interpretation of PREEMPT service action

5.6.10.4.2 Failed persistent reservation preempt

If the preempting I_T nexus' PREEMPT service action or PREEMPT AND ABORT service action fails (e.g., repeated TASK SET FULL status, repeated BUSY status, SCSI transport protocol time-out, or time-out due to the task set being blocked due to failed initiator port or failed SCSI initiator device), the application client may send a LOGICAL UNIT RESET task management function to the failing logical unit to remove blocking tasks and then reissue the preempting service action.

5.6.10.4.3 Preempting persistent reservations and registration handling

An application client may preempt the persistent reservation with another persistent reservation by issuing a PERSISTENT RESERVE OUT command with PREEMPT service action or PREEMPT AND ABORT service action through a registered I_T nexus with the following parameters:

a) RESERVATION KEY field set to the value of the reservation key that is registered with the logical unit for the I_T nexus;
b) SERVICE ACTION RESERVATION KEY field set to the value of the reservation key of the persistent reservation to be preempted; and
c) TYPE and SCOPE fields set to define a new persistent reservation. The SCOPE and TYPE of the persistent reservation created by the preempting I_T nexus may be different than those of the persistent reservation being preempted.

If the SERVICE ACTION RESERVATION KEY field identifies a persistent reservation holder (see 5.6.9), the device server shall perform a preempt by doing the following as an uninterrupted series of actions:

a) Release the persistent reservation for the holder identified by the SERVICE ACTION RESERVATION KEY field;
b) Remove the registrations for all I_T nexuses identified by the SERVICE ACTION RESERVATION KEY field, except the I_T nexus that is being used for the PERSISTENT RESERVE OUT command. If an all registrants persistent reservation is present and the SERVICE ACTION RESERVATION KEY field is set to zero, then all registrations shall be removed except for that of the I_T nexus that is being used for the PERSISTENT RESERVE OUT command;
c) Establish a persistent reservation for the preempting I_T nexus using the contents of the SCOPE and TYPE fields;
d) Process tasks as defined in 5.6.1;~~ and~~
e) Establish a unit attention condition for the initiator port associated with every I_T nexus that lost its persistent reservation and/or registration, with the additional sense code set to REGISTRATIONS PREEMPTED; and
f) If the TYPE or SCOPE have changed, then for every I_T nexus whose reservation key was not removed except for the I_T nexus on which the PERSISTENT RESERVE OUT command was received, the device server shall establish a unit attention condition for the initiator port associated with that I_T nexus, with the additional sense code set to RESERVATIONS RELEASED. If the TYPE or SCOPE have not changed, then no unit attention condition(s) shall be established for this reason.

After GOOD status has been returned for the PERSISTENT RESERVE OUT command, new tasks are subject to the persistent reservation restrictions established by the preempting I_T nexus.

The following tasks shall be subjected in a vendor specific manner either to the restrictions established by the persistent reservation being preempted or to the restrictions established by the preempting I_T nexus:

a) A task received after the arrival, but before the completion of the PERSISTENT RESERVE OUT command with the PREEMPT service action or the PREEMPT AND ABORT service action; or
b) A task in the dormant, blocked, or enabled state (see SAM-4) at the time the PERSISTENT RESERVE OUT command with the PREEMPT service action or the PREEMPT AND ABORT service action is received.

Completion status shall be returned for each task unless it was aborted by a PERSISTENT RESERVE OUT command with the PREEMPT AND ABORT service action and TAS bit set to zero in the Control mode page (see 7.4.6).

If an all registrants persistent reservation is not present, it is not an error for the persistent reservation holder to preempt itself (i.e., a PERSISTENT RESERVE OUT with a PREEMPT service action or a PREEMPT AND ABORT service action with the SERVICE ACTION RESERVATION KEY value equal to the persistent reservation holder's reservation key that is received from the persistent reservation holder). In that case, the device server shall establish the new persistent reservation and maintain the registration.

5.6.10.4.4 Removing registrations

When a registered reservation key does not identify a persistent reservation holder (see 5.6.9), an application client may remove the registration(s) without affecting any persistent reservations by issuing a PERSISTENT RESERVE OUT command with PREEMPT service action through a registered I_T nexus with the following parameters:

a) RESERVATION KEY field set to the value of the reservation key that is registered for the I_T nexus; and
b) SERVICE ACTION RESERVATION KEY field set to match the reservation key of the registration or registrations being removed.

If the SERVICE ACTION RESERVATION KEY field does not identify a persistent reservation holder or there is no persistent reservation holder (i.e., there is no persistent reservation), then the device server shall perform a preempt by doing the following in an uninterrupted series of actions:

a) Remove the registrations for all I_T nexuses specified by the SERVICE ACTION RESERVATION KEY field;
b) Ignore the contents of the SCOPE and TYPE fields;
c) Process tasks as defined in 5.6.1; and
d) Establish a unit attention condition for the initiator port associated with every I_T nexus that lost its registration other than the I_T nexus on which the PERSISTENT RESERVE OUT command was received, with the additional sense code set to REGISTRATIONS PREEMPTED.

If a PERSISTENT RESERVE OUT with a PREEMPT service action or a PREEMPT AND ABORT service action sets the SERVICE ACTION RESERVATION KEY field to a value that does not match any registered reservation key, then the device server shall return a RESERVATION CONFLICT status.

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 I_T nexus on which the PERSISTENT RESERVE OUT command was received. The registration is removed.

5.6.10.5 Preempting and aborting

The application client's request for and the device server's responses to a PERSISTENT RESERVE OUT command PREEMPT AND ABORT service action are identical to the responses to a PREEMPT service action (see 5.6.10.4) except for the additions described in this subclause. If no reservation conflict occurred, the device server shall perform the following uninterrupted series of actions:

a) If the persistent reservation is not an all registrants type then:
   A) If the TST field is 000b (see 7.4.6) and the faulted I_T nexus (see 3.1.37), if any, is not the I_T nexus associated with the persistent reservation or registration being preempted, then the task set ACA condition shall be processed as defined in SAM-4;
   B) If the TST field contains 000b and the faulted I_T nexus, if any, is the I_T nexus associated with the persistent reservation or registration being preempted, then the PERSISTENT RESERVE OUT command shall be processed without regard for the task set ACA condition; or
   C) If the TST field contains 001b, then the ACA condition shall be processed as defined in SAM-4;
b) Perform the uninterrupted series of actions described for the PREEMPT service action (see 5.6.10.4);

c) All tasks from the I_T nexus(es) associated with the persistent reservations or registrations being preempted (i.e., preempted tasks) except the task containing the PERSISTENT RESERVE OUT command itself shall be aborted as defined in SAM-4. If an aborted task is a command that causes the device server to generate additional commands and data transfers (e.g., EXTENDED COPY), then all commands and data transfers generated by the command shall be aborted before the ABORT TASK SET task management function is considered completed. After the ABORT TASK SET function has completed, all new tasks are subject to the persistent reservation restrictions established by the preempting I_T nexus;
d) If the persistent reservation is not an all registrants type, then the device server shall clear any ACA condition associated with an I_T nexus being preempted and shall abort any tasks with an ACA attribute received on that I_T nexus;
e) If the persistent reservation is an all registrants type, then:
   A) If the service action reservation key is set to zero, the device server shall clear any ACA condition and shall abort any tasks with an ACA attribute; or
   B) If the service action reservation key is not set to zero, the device server shall do the following for any I_T nexus registered using the specified reservation key:
      a) Clear any ACA condition; and
      b) Abort any tasks with an ACA attribute;

   and
f) For logical units that implement the PREVENT ALLOW MEDIUM REMOVAL command (see SBC-3, SSC-3, and SMC-3), the device server shall perform an action equivalent to the processing of a PREVENT ALLOW MEDIUM REMOVAL command with the PREVENT field equal to zero received on the I_T nexuses associated with the persistent reservation being preempted.

The actions described in this subclause shall be performed for all I_T nexuses that are registered with the non-zero SERVICE ACTION RESERVATION KEY value, without regard for whether the preempted I_T nexuses hold the persistent reservation. If the SERVICE ACTION RESERVATION KEY value is zero and an all registrants persistent reservation is present, the device server shall abort all tasks for all registered I_T nexuses.

5.6.10.6 Clearing

Any application client may release the persistent reservation and remove all registrations from a device server by issuing a PERSISTENT RESERVE OUT command with CLEAR service action through a registered I_T nexus with the following parameter:

a) RESERVATION KEY field set to the value of the reservation key that is registered with the logical unit for the I_T nexus.

In response to this request the device server shall perform a clear by doing the following as part of an uninterrupted series of actions:

a) Release the persistent reservation, if any;
b) Remove all registration(s) (see 5.6.6);
c) Ignore the contents of the SCOPE and TYPE fields;
d) Continue normal processing of any tasks from any I_T nexus that have been accepted by the device server as allowed (i.e., nonconflicting); and
e) Establish a unit attention condition for the initiator port associated with every registered I_T nexus other than the I_T nexus on which the PERSISTENT RESERVE OUT command with CLEAR service action was received, with the additional sense code set to RESERVATIONS PREEMPTED.

NOTE 1 - Application clients should not use the CLEAR service action except during recovery operations that are associated with a specific initiator port, since the effect of the CLEAR service action defeats the persistent reservations features that protect data integrity.