

# ENDL TEXAS

Date: 31 July 2004  
To: T10 Technical Committee  
From: Ralph O. Weber  
Subject: Review of SPC-3 r20 changes made during incorporation of 03-337r7

During the incorporation of 03-337r7, proposal tables 3 and 4 (concerning the REGISTER and REGISTER AND IGNORE EXISTING KEY service actions) were changed to match the format of the newer proposal table 5 (concerning the REGISTER AND MOVE service action).

In the process it was noticed that the ignore existing keys aspect of the REGISTER AND IGNORE EXISTING KEY service action was not properly represented.

While completing the incorporation, the following technical changes were made to bring the REGISTER AND IGNORE EXISTING KEY service action definition in line with previously applied definitions:

- The presence or absence of existing keys is ignored for all I\_T nexuses when the SPEC\_I\_PT bit is set to one (in 03-337r7, setting the SPEC\_I\_PT bit to one was undefined in the I\_T nexus on which the command was received was unregistered, but setting the SPEC\_I\_PT bit one resulted in a CHECK CONDITION if the I\_T nexus on which the command was received was registered).
- A new unit attention condition, RESERVATION KEY CHANGED, was defined and required to be used when a REGISTER AND IGNORE EXISTING KEY changes the reservation key of an I\_T nexus other than the I\_T nexus on which the command was received.

Because these changes are technical, CAP working group review is needed.

For convenience, the affected SPC-3 r20 pages are attached to this proposal in their original format with their original page numbers and headings.

In response to a PERSISTENT RESERVE IN command with READ RESERVATION service action the device server shall report the following information for the persistent reservation, if any:

- a) The current PRgeneration value (see 6.11.2);
- b) The registered reservation key, if any, associated with the I\_T nexus that holds the persistent reservation. If the persistent reservation is an all registrants type, the registered reservation key reported shall be zero; and
- c) The scope and type of the persistent reservation, if any.

If an application client uses a different reservation key for each I\_T nexus the application client may use the reservation key to associate the persistent reservation with the initiator port that holds the persistent reservation. This association is done using techniques that are outside the scope of this standard.

#### 5.6.5.4 Reporting full status

An application client may issue a PERSISTENT RESERVE IN command with READ FULL STATUS service action to receive all information about registrations and the persistent reservation, if any.

In response to a PERSISTENT RESERVE IN command with READ FULL STATUS service action the device server shall report the current PRgeneration value (see 6.11.2) and, for every I\_T nexus that is currently registered, the following information:

- a) The registered reservation key;
- b) Whether the I\_T nexus is a persistent reservation holder;
- c) If the I\_T nexus is a persistent reservation holder, the scope and type of the persistent reservation;
- d) The relative target port identifier identifying the target port of the I\_T nexus; and
- e) A TransportID identifying the initiator port of the I\_T nexus.

#### 5.6.6 Registering

To establish a persistent reservation the application client shall first register an I\_T nexus with a logical unit. An application client registers with a logical unit by issuing a PERSISTENT RESERVE OUT command with REGISTER service action or REGISTER AND IGNORE EXISTING KEY service action.

If an I\_T nexus has not yet established a reservation key or the reservation key and registration have been removed, an application client may register that I\_T nexus and zero or more specified unregistered I\_T nexuses by issuing a PERSISTENT RESERVE OUT command with REGISTER service action as defined in table 32.

If the I\_T nexus has an established registration the application client may change the reservation key by issuing a PERSISTENT RESERVE OUT command with REGISTER service action as defined in table 32.

**Table 32 — Register behaviors for a REGISTER service action**

Command I_T nexus status	Parameter list fields <sup>a</sup>			Results
	RESERVATION KEY	SERVICE ACTION RESERVATION KEY	SPEC_I_PT	
received on an unregistered I_T nexus	zero	zero	ignore	Do nothing except return GOOD status.
		non-zero	zero	Register the I_T nexus on which the command was received with the value specified in the SERVICE ACTION RESERVATION KEY field.
			one	Register the I_T nexus on which the command was received and each unregistered I_T nexus specified in the parameter list with the value specified in the SERVICE ACTION RESERVATION KEY field. <sup>b</sup>
	non-zero	ignore	ignore	Return RESERVATION CONFLICT status.
received on a registered I_T nexus	Not equal to I_T nexus reservation key	ignore	ignore	Return RESERVATION CONFLICT status.
	Equal to I_T nexus reservation key	zero	zero	Unregister the I_T nexus on which the command was received (see 5.6.10.3).
			one	Return CHECK CONDITION status. <sup>c</sup>
		non-zero	zero	Change the reservation key of the I_T nexus on which the command was received to the value specified in the SERVICE ACTION RESERVATION KEY field.
			one	Return CHECK CONDITION status. <sup>c</sup>

<sup>a</sup> For requirements regarding the parameter list fields not shown in this table see 6.12.3.

<sup>b</sup> If any I\_T nexus specified in the parameter list is registered, the command shall be terminated with a CHECK CONDITION status, the sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

<sup>c</sup> The sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

Alternatively, an application client may establish a reservation key for one or more I\_T nexuses without regard for whether one has previously been established by issuing a PERSISTENT RESERVE OUT command with REGISTER AND IGNORE EXISTING KEY service action as defined in table 33.

**Table 33 — Register behaviors for a REGISTER AND IGNORE EXISTING KEY service action**

Command I_T nexus status	Parameter list fields <sup>a</sup>		Results
	SERVICE ACTION RESERVATION KEY	SPEC_I_PT	
received on an unregistered I_T nexus	zero	ignore	Do nothing except return GOOD status.
	non-zero	zero	Register the I_T nexus on which the command was received with the value specified in the SERVICE ACTION RESERVATION KEY field.
		one	a) Register the I_T nexus on which the command was received with the value specified in the SERVICE ACTION RESERVATION KEY field; b) Register each unregistered I_T nexus specified in the parameter list with the value specified in the SERVICE ACTION RESERVATION KEY field; and c) Change the reservation key of each registered I_T nexus specified in the parameter list to the value specified in the SERVICE ACTION RESERVATION KEY field. <sup>c</sup>
received on a registered I_T nexus	zero	zero	Unregister the I_T nexus on which the command was received (see 5.6.10.3).
		one	Return CHECK CONDITION status. <sup>b</sup>
	non-zero	zero	Change the reservation key of the I_T nexus on which the command was received to the value specified in the SERVICE ACTION RESERVATION KEY field.
		one	a) Change the reservation key of the I_T nexus on which the command was received to the value specified in the SERVICE ACTION RESERVATION KEY field; b) Register each unregistered I_T nexus specified in the parameter list with the value specified in the SERVICE ACTION RESERVATION KEY field; and c) Change the reservation key of each registered I_T nexus specified in the parameter list to the value specified in the SERVICE ACTION RESERVATION KEY field. <sup>c</sup>

<sup>a</sup> The RESERVATION KEY field is ignored when processing a REGISTER AND IGNORE EXISTING KEY service action. For requirements regarding other parameter list fields not shown in this table see 6.12.3.

<sup>b</sup> The sense key shall be set to ILLEGAL REQUEST, and the additional sense code shall be set to INVALID FIELD IN CDB.

<sup>c</sup> A unit attention condition shall be established for the initiator port of each I\_T nexus that has its reservation key changed, with an additional sense code of RESERVATION KEY CHANGED.

Setting the SERVICE ACTION RESERVATION KEY field to zero in a PERSISTENT RESERVE OUT command with a REGISTER service action or REGISTER AND IGNORE EXISTING KEY service action causes the existing registration of the I\_T nexus that sent the command to be removed (i.e., the I\_T nexus is unregistered) as shown in table 32 and table 33, respectively.

If a PERSISTENT RESERVE OUT command with a REGISTER service action or a REGISTER AND IGNORE EXISTING KEY 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 code shall be set to INSUFFICIENT REGISTRATION RESOURCES.

In response to a PERSISTENT RESERVE OUT command with a REGISTER service action or a REGISTER AND IGNORE EXISTING KEY service action the device server shall perform a registration for each specified I\_T nexus 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 I\_T nexus being registered using the indication of the target port associated with the registration and either the initiator port name (see 3.1.49) on SCSI transport protocols where port names are required or the initiator port identifier (see 3.1.48) on SCSI transport protocols where port names are not required;
- e) Register the reservation key without changing any 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 I\_T nexus to be processed. The device server shall retain the reservation key until the key is changed as described in this subclause or removed as described in 5.6.10.

Any PERSISTENT RESERVE OUT command service action received from an unregistered I\_T nexus, other than the REGISTER or the REGISTER AND IGNORE EXISTING KEY service action, shall be rejected with a RESERVATION CONFLICT status.

It is not an error for an I\_T nexus that is registered to be registered again with the same reservation key or a new reservation key. A registration shall have no effect on any other registrations (e.g., when more than one I\_T nexus is registered with the same reservation key and one of those I\_T nexuses registers again it has no effect on the other I\_T nexus' registrations). A registration that contains a non-zero value in the SERVICE ACTION RESERVATION KEY field shall have no effect on any persistent reservations (i.e., the reservation key for an I\_T nexus may be changed without affecting any previously created persistent reservation).

Multiple I\_T nexuses may be registered with the same reservation key. An application client may use the same reservation key for other I\_T nexuses and logical units.