Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94045
415 960-1300

April 18, 1998

% Sun Logo for

John Lohmeyer
Chairperson, X3T10
Symbios Logic Inc.
1635 Aeroplaza Drive
Colorado Springs, Colorado 80916

Subject: Proposal for Persistent Reservation

This proposal extends the definition of reservations to allow proper behavior in multi-initiator and multi-port environments. The proposal defines persistent reservations which remain valid across Target Reset and can only be cleared by power down or by a properly qualified persistent reservation from another initiator. Using the commands defined by this proposal, a host can protect the logical unit from improper behavior caused by another initiator on the same or other ports. At the same time, the host can determine from a logical unit which initiators share the logical unit, which initiator is presently reserving the logical unit, and can choose to displace the reservation of an initiator which is known to have failed.

Sincerely,

Robert N. Snively
Sun Microsystems
Mail Stop MPK 12-204
2550 Garcia Ave.
Mountain View, CA 94043-1100
phone: 415-786-6694
e-mail: bob.snively@sun.com

Change control

Changes from Revision 0 to Revision 1 (results of X3T10 meeting, June 1995)

1) Clear Action removed
False reference to Clear Key action removed, SPC, section N.n.1.3

2) PROUT Parameters clarified
Table 6 and related text clarified

3) “Shallify” all sections
All portions of the document stating requirements are modified to use the word “shall” to indicate a requirement.

4) Clarify PERSISTENT RESERVE IN Read Key parameters
Table 10 text is modified to more clearly identify the first reservation key.

5) Generation value initialization
SPC, Section N.m.2.1, is modified to indicate that the Generation value is initialized to a value of zero during the power on reset processing.

6) Clarify PERSISTENT RESERVE IN Read Reservations parameters
Table 11 text is modified to more clearly identify the first reservation key.

7) ASC/ASCQ definitions
The errors are interpreted as standard SCSI ASC/ASCQ codes where possible.

8) Include in SMC command set
The PROUT and PRIN commands are to be included in SMC as well as SBC, SSC, and SCC.

9) Operation codes added
PROUT is 5fh and PRIN is 5Eh.

10) Mandatory definition modified
The commands will be defined as selectively mandatory. The text for the notes for SBC, SSC, SCC, and SMC will be completed.

11) Description clarified
The text of the third paragraph of the description, page 4, is rewritten to be more clear.

12) Text corrected
The text of the third paragraph of section 5.3 for SPC is grammatically corrected.

13) Low level contention resolution allowed
The text of SPC section 5.4, paragraph 2 is modified to allow resolution of contention at the link level, while still allowing the higher level management of contention.

14) Task management cases clarified and LOGICAL UNIT RESET included.
The text of SPC section 5.4, last indented section, is modified to clarify that task management may only apply to a certain logical unit or to all logical units in a target. In addition, text for LOGICAL UNIT RESET is included on the assumption that the function will be approved.

n

15) Setting of Reservation Key
The text is modified in sections N.n.1.2 of SPC to meet the intent that the key can only be modified by a Register action. Improper modifications of the key are RESERVATION CONFLICTS. Note that the reservation key is now checked for the Register action.

16) Task attributes
Task attributes of ordered are recommended for the Reserve action. The behavior of the other actions does not require any particular task attribute.

17) Conflicts between PROUT and RESERVE
Reservations created by RESERVE and persistent reservations created by PROUT to the same LUN will always conflict. This is already expressed in the text to be placed in SPC, but had to be included in the overall description.

18) Invalid reservation key
An invalid reservation key generates a CHECK CONDITION and an ASC/ASCQ of Invalid Parameter. This needs to be corrected in a number of places.

19) Element Reservation
An element reservation scope is defined for use by the SMC command set. This is structured in a manner similar to the Extent scope, except that the qualifying parameters are the two byte element address instead.

20) Scope, optional and mandatory
The LUN scope is mandatory for the commands, but the element and extent scopes are optional. Appropriate error codes are defined.

21) Power Cycle resets reservation key
The text is modified to clarify that the reservation key is set to its default value of 0 by power-on reset.

Persistent Reservation Proposal

Description of function / Persistent Reservation model:

Two new commands are defined in SPC, PERSISTENT RESERVE IN (PRIN) and PERSISTENT RESERVE OUT (PROUT). The commands are used to create and release persistent reservations (including logical unit or Extent reservations), to provide reservation keys to logical units, and to force actions on tasks from other initiators.

The PRIN and PROUT commands have the following capabilities, many of which are optional:

- Create persistent reservations using PROUT.
  - Extent (shared reservations defined)
  - Exclusive logical unit
  - Shared logical unit (shared reservations defined)
- Release persistent reservations using PROUT.
- Register an 8-byte Reservation Key for the sourcing initiator with the attached device using PROUT.
- Determine the present reservation's key and characteristics using PRIN.
- Determine the key of other initiators that are attached to the peripheral device using PRIN.
- Preempt a reservation with another initiator with new reservation from this using PROUT.
- Optionally automatically clear all tasks related with the preempted reservation.
- Generate a 32-bit number increasing with each reservation to warn of intervening actions.

A persistent reservation is formed between an initiator and a logical unit when the PROUT command containing the parameters requesting a persistent reservation is successfully executed. The parameters indicate the type of reservation that is formed. In addition, the parameters contain an 8-byte reservation key that is used by the target and by software on other initiators to identify the initiator holding the reservation. The persistent reservation cannot be released by a Target Reset, other reset activity, or by a RELEASE command. It can be released only by the reserving initiator using a PROUT command containing the release parameters, by a power off, or by another initiator using the PROUT command containing the proper preemptive reservation parameters and the reservation key of the initiator holding the reservation. While the persistent reservation is active, any conflicting persistent reservation or activity that conflicts with the reservation is rejected with RESERVATION CONFLICT status. This behavior allows any cooperating initiators having access to the logical unit through any port to execute carefully managed reservation protocols that will allow the logical unit to be safely shared among them. The reservation is safe from interruption by any initiators not participating in the persistent reservation protocol, even during booting and error recovery operations.

The use of a reservation key as part of the reservation process enables initiators to identify other ports holding reservations or sharing the logical unit. Hosts use the reservation key to perform locking and failover recovery operations. The communication for such algorithms is usually through an auxiliary port such as Ethernet or Fibre Channel. Targets with reservations held by failing hosts can be identified and preemptively reserved by hosts that are still

operational. When preemptive reservations are performed, the reserving initiator can optionally invoke an automatic clearing of all tasks for the initiator port that is being preempted.

The reservation key of other ports that have previously generated persistent reservations can be obtained from the logical unit to allow each initiator to monitor which initiator ports have participated in the sharing process.

The programming conventions typically used with RESERVE and RELEASE conflict with the programming conventions that are used with PROUT and PRIN. Operating systems should only use one of the two reservation command sets at a time within a single logical unit. Reservations established by RESERVE are defined as conflicting with those defined by PROUT and vice versa.

Modifications required to SPC to implement persistent reservation:

1) Provide additional description in section 5.3 of SPC

Section 5.3 and 5.3.1 should be rewritten as follows:

5.3 Reservations

Various types of reservation commands can be used to prohibit or restrict the execution of certain commands to a logical unit or a portion of the logical unit. Using these reservation commands, application clients can cooperate to protect shared data from accidental modification. If the application clients do not cooperate in the execution of a reservation protocol, data may be unexpectedly modified and deadlock conditions may occur.

Two types of reservation commands are defined. The nonpersistent reservation commands, RESERVE(6), RESERVE(10), RELEASE(6), and RELEASE(10) are used among multiple initiators that do not require operations to be protected across initiator failures. The reservations created by such commands include reservations for the entire logical unit or for an extent of the logical unit. Extent reservations may place restrictions only upon certain types of commands. The reservations may also be made restricting access to the device to a different initiator, usually a temporary initiator performing a service for the reserving initiator. The reservations do not persist across some recovery actions, so most systems using nonpersistent reservations require significant reinitialization after a failure. Reservations are retained by the logical unit until released or until reset by mechanisms specified in this standard.

The persistent reservation commands, PERSISTENT RESERVE IN and PERSISTENT RESERVE OUT, are used among multiple initiators that require operations to be protected across initiator failures. The reservations created by such commands include reservations for the entire logical unit or for an extent of the logical unit. Both extent and logical unit reservations may place restrictions on certain types of commands. The reservations do persist across recovery actions, so that recovery can be managed without requiring complete reinitialization of the system. Reservations for failing initiators can be preempted by an initiator as part of the

recovery process. Reservations are retained by the logical unit until released or until reset by mechanisms specified in this standard.

Because a device server cannot differentiate among different application clients running on an initiator, all application clients on the initiator have the same access restrictions. When multiple application clients are accessing a single device server from one initiator, the application clients shall coordinate reservations.

The clause defining each command's operations shall contain a description of how that command is affected by reservations. The command may be allowed to execute or it may be prevented from execution. If the command is prevented from execution, the command is said to conflict with the reservation and the logical unit shall present a status of RESERVATION CONFLICT. Commands that read or write to the storage medium or to storage caches shall obey the rules defined in the clauses describing the RESERVE command and the PERSISTENT RESERVE OUT command. Commands that retrieve or alter information about the device server's operating state shall conflict with logical unit reservations unless otherwise specified. Commands that alter information about the device server's operating state shall conflict, unless otherwise specified, with extent reservations unless the logical unit maintains separate state information for each initiator.

The INQUIRY and REQUEST SENSE commands shall not be affected by any kind of reservation. The RESERVE(6) and RESERVE(10) commands allow superceding reservations and shall be executed even when nonpersistent reservations are present from the same initiator. The RELEASE(6) and RELEASE(10) commands shall be executed for the reserving initiator even when nonpersistent reservations are present.

The execution of a RESERVE(6) or a RESERVE(10) command conflicts with a persistent reservation. The execution of a PERSISTENT RESERVE OUT command with a Reserve, Release, Preempt, or Preempt and Clear action conflicts with a nonpersistent reservation.

2) Replace section 5.4 of SPC concerning Dual Port behavior

The entire clause entitled Dual Port behavior should be replaced with the following text:

5.4 Multiple port and multiple initiator behavior

The SCSI Architectural Model, X3.XXX-19xx, specifies the behavior of logical units being accessed by more than one initiator. Additional ports to a logical unit provide alternate delivery paths through which the device server can be reached and may also provide connectivity for additional initiators. An alternate path can be used to improve the availability of drives in the presence of certain types of failures and to improve the performance of drives whose other paths may be busy.

If a logical unit has more than one SCSI interface port, the arbitration and connection management among the ports is defined by the interconnect implementation. Above the interconnect implementation, two contention resolution options exist:

1) If one port to a logical unit is being used by an initiator, accesses attempted through another port may receive a status of BUSY.

2) If the logical unit has sufficient internal resources, the logical unit may accept actions through other ports while one port is being used.

The device server shall indicate the presence of multiple ports by setting the DualP bit to *1* in its standard INQUIRY data.

From an initiator, each other initiator attached to the logical unit has the same relationship to the logical unit, whether the other initiator accesses the logical unit through the same or a different port. Reservations, persistent reservations, and task management functions are performed between a single initiator and a single logical unit. The following operations are the only operations that allow an initiator to interact with the tasks of another initiator:

PERSISTENT RESERVE OUT with Preempt action removes persistent reservations for another initiator.

PERSISTENT RESERVE OUT with Preempt and Clear action removes persistent reservations and all tasks for another initiator.

Task Management function of TARGET RESET removes nonpersistent reservations and removes all tasks for all logical units of the target and for all initiators. Persistent reservations remain unmodified.

Task Management function of LOGICAL UNIT RESET removes nonpersistent reservations and removes all tasks for all initiators for the addressed logical unit and any logical units depending from it in a hierarchical addressing structure. Persistent reservations remain unmodified. [Editor's note: This paragraph is included in expectation of the acceptance of the LOGICAL UNIT RESET task management function.]

Task Management function of CLEAR TASK SET removes all tasks for the selected logical unit for all initiators. Most other machine states remain unmodified, including MODE SELECT PARAMETERS, persistent and nonpersistent reservations, and auto contingent allegiance.

3) Remove reference to the Port Status command in Table 5

At present, the referenced section entitled "Port Status" is not included in revision 6 of SPC. It should continue to be not included and should be removed from table 5, page 16.

4) Include PROUT and PRIN commands in SPC:

The PERSISTENT RESERVE OUT command and the PERSISTENT RESERVE IN command will be included in table 5 as "Z" commands.

The following text will be added to SPC:

N.n PERSISTENT RESERVE OUT command

The PERSISTENT RESERVE OUT or PROUT command (see Table 1) is used to reserve a logical unit or an extent within a logical unit for the exclusive or shared use of a particular initiator. The command is used in conjunction with the PERSISTENT RESERVE IN command and cannot be used with the RESERVE and RELEASE commands. Persistent reservations conflict with reservations established by the RESERVE command. Initiators performing PROUT actions are identified by a reservation key provided by the application client. An application client can use the PRIN command to identify which applications are holding conflicting or invalid reservations and use the PROUT command to preempt those reservations if required.

The PROUT and PRIN commands provide the basic mechanism for dynamic contention resolution in multiple-initiator systems using multiple port targets. The identification of reservations using the reservation key makes it possible to determine which ports hold conflicting reservations and to take over reservations from failing or uncooperative initiators.

Table 1: PERSISTENT RESERVE OUT command

Bits / Bytes	7	6	5	4	3	2	1	0
0	Operation Code (5Fh)							
1	reserved				Action			
2	Scope				Type			
3	reserved							
4	reserved							
5	reserved							
6	reserved							
7	(MSB) Parameter Length							
8	Parameter Length (LSB)							
9	Control							

The PROUT command contains fields that specify a persistent reservation action, the intended scope of the reservation, and the restrictions caused by the reservation. Parameters contained in the PROUT Parameters specify the reservation keys and extent information required to perform a particular persistent reservation action. The parameters are 24 bytes in length and the Parameter Length field is required to have a value of 24. If the Parameter Length is not 24, a CHECK CONDITION is indicated with an ASC/ASCQ of Invalid Parameter List Length (1A/00). Since persistent reservations are not reset by Target Reset or other global actions, they can be used to enforce device sharing among multiple initiators.

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 persistent reservation:

INQUIRY

REQUEST SENSE

PREVENT ALLOW MEDIUM REMOVAL (with a prevent bit of one)

PERSISTENT RESERVE IN

PERSISTENT RESERVE OUT (with an action of Preempt)

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

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

Other commands conflict if they perform an operation to the logical unit that violates either the scope or the type specified for an active persistent reservation. Each command of a set of linked commands is individually examined for conflicts at the time the command is received by the target.

N.n.1 PROUT Reservation Actions:

The PROUT command actions are shown in Table 2. The parameters required for each action are shown in Table 6.

Table 2: PERSISTENT RESERVE OUT Action

Action Code (Hex)	Action Name	Action Description
00	Register	Register Reservation Key with target
01	Reserve	Create Persistent Reservation using Reservation Key
02	Release	Release Persistent Reservation
03	(reserved)	
04	Preempt	Pre-empt Persistent Reservation for other initiator
05	Pre-empt and Clear	Persistent Reservation and Clear Task Set for preempted initiator
06-1F	(reserved)	

N.n.1.1 Register

The PROUT command executing a Register action registers a reservation key with a target without generating a reservation. For each initiator that performs a PROUT Register action, the target retains the reservation key until the key is changed by a new PROUT command with the

Register action from the same initiator or until the key is reset to the default value of “0” by powering down the target. The Register action can be performed regardless of any active persistent reservations. All existing persistent reservations for the initiator receive the new reservation key. If the present reservation key is not correctly provided by the parameters of the Register action, a status value of CHECK CONDITION with an ASC/ASCQ of Invalid Parameter (29/00) shall be generated when the Register action is attempted.

For predictable behavior, there are no task attribute requirements for the Register action.

N.n.1.2 Reserve

The PROUT command performing a Reserve action creates a persistent reservation having a specified scope and type. The scope and type of a persistent reservation are defined below. A status of CHECK CONDITION and an ASC/ASCQ of Invalid Parameter (29/00) shall be generated for a PROUT command that specifies a reservation key other than the key for the initiator.

A status of RESERVATION CONFLICT shall be generated for a PROUT command that specifies the execution of a Reserve action that conflicts with any active persistent reservations from the same initiator in scope, type, extent, or reservation key at the time the PROUT is enabled for execution. For the simplest predictable behavior, the Reserve action should be performed with the Ordered task attribute.

Persistent reservations shall not be superceded by a new persistent reservation from any initiator except by execution of a PROUT specifying either the Preempt or Preempt and Clear action. New persistent reservations that do not conflict with an existing persistent reservation shall be executed normally. The reservation of a logical unit or the reservation of reserved extents having the same type value is permitted if no conflicting persistent reservations are held by another initiator. When such overlapping reservations are released, each of the extent reservations and the logical unit reservation shall be removed with a separate Release action.

A PROUT command not performing a Preempt or a Preempt and Clear action shall not be performed and shall be ended with status of RESERVATION CONFLICT if there are any queued or active tasks from any initiator that would conflict with the reservation to be established at the time the PROUT command is enabled for execution.

N.n.1.3 Release

The PROUT command performing a Release action removes an active persistent reservation held by the same initiator. The parameters associated with the Release action must match the parameters of the active reservation. It is not an error to send a PROUT specifying a Release action when no persistent reservation exists from that initiator. The reservation key is not changed by the Release action.

A status of CHECK CONDITION and an ASC/ASCQ of Invalid Release of Active Persistent Reservation shall be generated for a PROUT command that specifies the release of a persistent reservation with an incorrect scope, reservation key, or extent.

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

1) Power off. The power off also performs a hard reset and sets the reservation key to its default value of 0.

2) Execution of a PROUT command from another initiator with a Persistent Reserve Action of Preempt or Preempt and Clear

For predictable behavior, there are no task attribute requirements for the Release action.

N.n.1.4 Preempt

The PROUT command performing a Preempt action removes all reservations for the initiator specified by the PROUT parameter page. The initiator is identified by the reservation key of the initiator to be preempted. Any commands from any initiator that have been accepted by the logical unit as nonconflicting will continue normal execution.

A Unit Attention condition is established for the preempted initiator. The first new command from the preempted initiator shall present CHECK CONDITION and present the Unit Attention condition with an ASC/ASCQ of Unit Attention/Reservations preempted. Subsequent commands are subject to the reservation restrictions established by the preempting initiator.

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PROUT command and the corresponding parameters in the PROUT parameter page.

For predictable behavior, there are no task attribute requirements for the Preempt action.

N.n.1.5 Preempt and Clear

The PROUT command performing a Preempt and Clear action removes all reservations for the initiator specified by the PROUT parameter page. The initiator is identified by the reservation key of the initiator to be preempted. Any commands from the initiator being preempted are each terminated as if an ABORT TASK task management function had been performed by the preempted initiator.

A Unit Attention condition is established for the preempted initiator. The first new command from the preempted initiator shall present CHECK CONDITION and present the Unit Attention condition with an ASC/ASCQ of Unit Attention/Reservations preempted. Subsequent new commands and retries of commands that timed out because they were cleared are subject to the reservation restrictions established by the preempting initiator.

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PROUT command and the corresponding parameters in the PROUT parameter page.

For predictable behavior, there are no task attribute requirements for the Preempt and Clear action.

N.n.2 PERSISTENT RESERVE OUT Scope

The value in the Scope field indicates whether a persistent reservation applies to an entire logical unit, to a portion of the logical unit defined as an extent, or to an element. If a Scope field specifies a scope that is not implemented, a status of CHECK CONDITION with an ASC/ASCQ value of Invalid CDB (24/00) is presented. The values of the Scope field are defined in Table 3.

Table 3: PERSISTENT RESERVE OUT Scope

Code (Hex)	Scope Name	Scope Description
0	LU	The Persistent Reservation Out is applied to the full logical unit.
1	Extent	The Persistent Reservation Out is applied to the specified extent.
2	Element	The Persistent Reservation Out is applied to the specified element.
3-F	(reserved)	

N.n.2.1 LU

A Scope field value of LU indicates that the persistent reservation applies to the entire logical unit. The LU scope shall be implemented by all targets that implement PROUT.

N.n.2.2 Extent

A Scope field value of Extent indicates that the persistent reservation applies to the extent of the logical unit defined by the extent parameters in the PROUT parameter page. An extent is defined only for devices defining contiguous logical block addresses. The Extent scope is optional for all targets that implement PROUT. The number of extents that can be reserved for a logical unit is implementation dependent.

N.n.2.3 Element

A Scope field value of Element indicates that the persistent reservation applies to the element of the logical unit defined by the element parameters in the PROUT parameter page. An element is defined by the Medium Changer Command Set (SMC). The Element scope is optional for all targets that implement PROUT.

N.n.3 PERSISTENT RESERVE OUT Type

The value in the Type field specifies the characteristics of the persistent reservation being established for all data blocks within the extent or within the logical unit. Table 4 describes the characteristics of the five different type values.

Table 4PERSISTENT RESERVE OUT Type

Code (Hex)	Type Name	Type Description		
		Read	Write	Permitted Reservations
0	Read Shared	Shared	Prohibited	Nonconflicting, any initiator
1	Write Exclusive	Shared	Exclusive	Nonconflicting, any initiator
2	Read Exclusive	Exclusive	Shared	Nonconflicting, any initiator
3	Exclusive Access	Exclusive	Exclusive	Nonconflicting, this initiator
4	Shared Access	Shared	Shared	Nonconflicting, this initiator
5-F	(reserved)			

N.n.3.1 Read

If a type allows read commands to be shared, then any initiator can execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator. If a type requires read commands to be exclusive, then only the reserving initiator can perform a transfer from the storage medium or cache to the initiator. Any SCSI commands from another initiator that would create such a transfer will receive status of RESERVATION CONFLICT.

N.n.3.2 Write

If a type allows write commands to be shared, then any initiator can execute commands that perform transfers from the initiator to the storage medium or cache of the logical unit. If a type requires write commands to be exclusive, then only the reserving initiator can perform a transfer from the initiator to the storage medium or cache of the logical unit. Any SCSI commands from another initiator that would create such a transfer will receive status of RESERVATION CONFLICT. If a type requires write commands to be prohibited, then any command from any initiator that would create such a transfer will receive status of RESERVATION CONFLICT.

N.n.3.3 Permitted Reservations

If a type allows nonconflicting reservations from any initiator, then any initiator may generate persistent reservations to the same extent or logical unit as long as the characteristics are not contrary to the existing reservations. Attempts to create conflicting reservations will receive status of RESERVATION CONFLICT.

If a type allows nonconflicting reservations only from the reserving initiator, then all attempts by other initiators to perform any PROUT command with the Reserve action will receive status of RESERVATION CONFLICT. Attempts to create conflicting reservations by the reserving initiator will receive status of RESERVATION CONFLICT.

N.n.4 PERSISTENT RESERVE OUT parameters

The parameters required to perform the PROUT command are defined in Table 5 and in Table 7. Table 6 indicates which parameters shall be set by the initiator and expected by the target for each Action and Scope value. The locations for all parameter fields shall be transmitted, even if the parameter is not required for the specified function

Table 5: PERSISTENT RESERVE OUT parameters

Bit	Function of Byte							
Byte	7	6	5	4	3	2	1	0
0 through 23	PROUT parameter page							

Table 6: PERSISTENT RESERVE OUT: Allowed actions and valid parameters

Action	Allowed Scope	Parameters Set by initiator/Expected by target		
		Key for port performing PROUT	Key for port changed by PROUT	Extent or Element Parameters
Register	LU	Set/Expected	Set/Expected	Not Set/Ignored
Reserve	LU	Set/Expected	Not Set/Ignored	Not Set/Ignored
Reserve	Extent	Set/Expected	Not Set/Ignored	Extent Set/Expected
Reserve	Element	Set/Expected	Not Set/Ignored	Element Set/Expected
Release	LU	Set/Expected	Not Set/Ignored	Not Set/Ignored
Release	Extent	Set/Expected	Not Set/Ignored	Extent Set/Expected
Release	Element	Set/Expected	Not Set/Ignored	Element Set/Expected
Preempt	LU	Set/Expected	Set/Expected	Not Set/Ignored
Preempt	Extent	Set/Expected	Set/Expected	Extent Set/Expected
Preempt	Element	Set/Expected	Set/Expected	Element Set/Expected
Preempt and Clear	LU	Set/Expected	Set/Expected	Not Set/Ignored
Preempt and Clear	Extent	Set/Expected	Set/Expected	Extent Set/Expected
Preempt and Clear	Element	Set/Expected	Set/Expected	Element Set/Expected

N.n.4.1 Reservation Key of initiator performing command

The reservation key is an 8-byte token provided by the initiator to the logical unit to identify the source of the PROUT command. The default value of the reservation key is set by power on reset and is 0.

N.n.4.2 Reservation Key of initiator being changed

The reservation key of the initiator being changed is used for three actions, the Register action and the two Preempt actions.

For the Register action, this reservation key is the new reservation key to be registered.

For the Preempt actions, this reservation key matches the reservation key of the persistent reservation that is being preempted. If it does not match, the PROUT command presents status of RESERVATION CONFLICT.

N.n.4.3 Extent definition parameters

The extent parameters are only required for those PROUT actions that use a Scope value of Extent. The extent is defined by the 32-bit starting logical block address and the 16-bit count of logical blocks in the extent.

N.n.4.4 Element definition parameters

The element address parameter is only required for those PROUT actions that use a Scope value of Element. The element address is provided in bytes 18 and 19.

Table 7: PERSISTENT RESERVE OUT parameter page

	Function of Byte							
Bit Byte	7	6	5	4	3	2	1	0
0 through 7	(MSB) Reservation Key of initiator performing command (LSB)							
8 through 15	(MSB) Reservation Key of initiator being changed (LSB)							
16 -19	(MSB) LBA of first block of Extent / Element Address (LSB)							
20	(reserved)							
21	(reserved)							
22 - 23	(MSB) Extent Length (MSB)							

N.m PERSISTENT RESERVE IN command

The PERSISTENT RESERVE IN or PRIN command (see Table 8) is used to obtain information about reservations and reservation keys that are active within a logical unit. The command is used in conjunction with the PERSISTENT RESERVE OUT command and cannot be used with the RESERVE and RELEASE commands.

The actual length of the parameters that could be returned by the PRIN command is defined within the parameter list. The Parameter Length field in the CDB indicates how much space has been reserved for the returned parameter list. If the length is not

sufficient to contain the entire parameter list, the first portion of the list is returned. If the remainder of the list is required, a new PRIN command with a Parameter Length field large enough to contain the entire list is executed.

For predictable behavior, there are no task attribute requirements for the PRIN command.

Table 8: PERSISTENT RESERVE IN command

Bits / Bytes	7	6	5	4	3	2	1	0
0	Operation Code (5Eh)							
1	reserved			Action				
2	reserved							
3	reserved							
4	reserved							
5	reserved							
6	reserved							
7	(MSB) Parameter Length							
8	(LSB)							
9	Control							

N.m.1 PERSISTENT RESERVE IN action codes

The action codes for the PRIN command are defined in Table 9.

Table 9: PERSISTENT RESERVE IN Action

Action Code (Hex)	Action Name	Action Description
00	Read Keys	Reads all registered Reservation Keys
01	Read Reservations	Reads all current reservations
02-1F	(reserved)	(reserved)

N.m.1.1 Read Keys

The Read Keys action value requests that the logical unit return a parameter list containing a header and a complete list of all reservation keys that have been passed to the logical unit from all initiators. The keys may have been passed by a PROUT command that has performed a reserve action, a register action, or one of the preempt actions. The reservation keys do not indicate what initiator or port is associated with the key. That information must be obtained from other initiators by mechanisms outside the scope of this standard.

N.m.1.2 Read Reservations

The Read Reservations action value requests that the logical unit return a parameter list containing a header and a complete list of all reservations that are presently active on the logical unit and its extents.

N.m.2 PERSISTENT RESERVE IN parameters for Read Keys

The format for the parameters provided in response to a PRIN command with the Read Keys action is shown in Table 10.

Table 10: PERSISTENT RESERVE IN parameters for Read Keys

	Function of Byte							
Bit / Byte	7	6	5	4	3	2	1	0
0 through 3	(MSB) Generation (LSB)							
4 through 7	(MSB) Additional length of parameter field (LSB)							
8 through 15	(MSB) First Reservation Key (LSB)							
8n through 7+8n	(MSB) Nth Reservation Key (LSB)							

N.m.2.1 Generation

The Generation value is a 32-bit counter in the logical unit that is incremented every time a PROUT command requests a Reserve, a Preempt, or a Preempt and Clear operation. The counter is not incremented by a PRIN command, by a PROUT command that performs a Register or Release action, or by a PROUT command that is not

n

performed due to an error or reservation conflict. The Generation value is set to 0 as part of the power on reset process.

The Generation value allows an initiator to verify that reservations have not been updated in the logical unit without the knowledge of the initiator.

N.m.2.2 Additional Length

This field contains a count of the number of bytes in the reservation key list. The total number of bytes available in a particular PRIN command is 8 more than the value in the Additional Length field. If the Parameter Length specified by the PRIN command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed Parameter Length are transmitted to the initiator. The remaining bytes are truncated, although the Additional Length field still contains the number of bytes in the reservation key list that would have been transmitted. In the case of such a truncation, CHECK CONDITION status with sense information of incorrect length is provided.

N.m.2.3 Reservation Key list

This list contains all the 8-byte reservation keys known to the logical unit through PROUT Reserve, Preempt, Preempt and Clear, or Register actions. Each reservation key may be examined by the host system and correlated with a particular initiator and SCSI port by mechanisms outside the scope of this standard.

N.m.3 PERSISTENT RESERVE IN parameters for Read Reservations

The format for the parameters provided in response to a PRIN command with the Read Reservations action is shown in Table 11.

Table 11: PERSISTENT RESERVE IN parameters for Read Reservations

	Function of Byte							
Bit / Byte	7	6	5	4	3	2	1	0
0 through 3	(MSB) Generation (LSB)							
4 through 7	(MSB) Additional length of parameter field (LSB)							
8 through 23	(MSB) First Read Reservation Page (LSB)							
-8 +16n through 7 +16n	(MSB) Nth Read Reservation Page (LSB)							

The Generation field and Additional Length fields are defined exactly the same as the fields for the parameters for the Read Keys action. The read reservation pages are defined in Table 12. There is one read reservation page for each reservation held on the logical unit by any initiator.

Table 12: PERSISTENT RESERVE IN Read Reservation Page

	Function of Byte							
Bit / Byte	7	6	5	4	3	2	1	0
0 through 7	(MSB) Reservation Key of port holding reservation (LSB)							
8 through 11	(MSB) LBA of first block of Extent / Element Address (LSB)							
12	(reserved)							
13	Scope				Type			
14 15	(MSB) Extent Length (MSB)							

N.m.3.1 Read Reservation Page parameters

For each reservation held on the logical unit, there shall be a Read Reservation page presented in the list of parameters returned by the logical unit to the PRIN command with a Read Reservations action. The page contains the reservation key of the initiator holding the reservation. The type and scope of the reservation are also defined. If the scope is an Extent reservation, then the LBA of the first block of the extent and the extent length parameters are valid. If the scope is an Element reservation, the the Element address is placed in bytes 10 and 11 and bytes 8, 9, 14, and 15 are set to 0. If the scope is a Logical Unit reservation, then bytes 8 through 11, 14, and 15 are set to 0.

New ASC/ASCQ definitions required:

The following new ASC/ASCQ indications or interpretations are required to present error information associated with persistent reservations:

Invalid CDB (24/00) (Invalid Release of Active Persistent Reservation)

Invalid Parameter (26/00) (offered when more than one parameter page is specified by PROUT Preempt).

Invalid Parameter (26/00) (offered when duplicate keys are established by a Register action)

Unit Attention/Reservations Preempted (Sense Code = 06, 2A/TBD)

Incorrect Length (ILI bit = 1, Sense Code = 0) (offered when Parameter Length is too short for PRIN parameter lists.)

Invalid Parameter List Length (1A/00) (offered when Parameter Length is not = 24 in PROUT)

Proposed text changes to SBC, SSC, SMC, and SCC

The PERSISTENT RESERVE IN and PERSISTENT RESERVE OUT commands shall be included in the introductory command maps for all four command sets.

PERSISTENT RESERVE IN, PERSISTENT RESERVE OUT, RESERVE(6), RESERVE(10), RELEASE(6), and RELEASE(10) shall be marked as S (for selectively mandatory) instead of M.

The following note shall be applied for RESERVE(6), RELEASE(6), RESERVE(10), RELEASE(10), PERSISTENT RESERVE IN, and PERSISTENT RESERVE OUT:

At least one pair of the following pairs of commands is mandatory:

RESERVE(6), RELEASE(6)

RESERVE(10), RELEASE(10)

PERSISTENT RESERVE IN, PERSISTENT RESERVE OUT

PERSISTENT RESERVE IN and PERSISTENT RESERVE OUT should be included in the command set of multi-ported devices.
If extended logical unit addresses are implemented, at least one pair of the following pairs of commands is mandatory:

RESERVE(10), RELEASE(10)

PERSISTENT RESERVE IN, PERSISTENT RESERVE OUT

Proposed text changes to other SCSI-3 documents:

None.