

Date: July 7, 1998

To: T10 Committee (SCSI)

From: George Penokie (IBM)

Subject: Persistent Reservations

### **0.0.1 The Persistent Reservations management method**

The Persistent Reservations management method is used among multiple initiators that require operations to be protected across initiator failures, which usually involve hard resets. Persistent reservations persist across recovery actions, to provide initiators with more detailed control over reservations recovery. Persistent reservations for failing initiators may be preempted or replaced by another initiator as part of the recovery process. Persistent reservations are retained by the device server until released, preempted, replaced, or until cleared by mechanisms specified in this standard. Persistent reservations are optionally retained when power to the target is lost.

#### **0.0.1.1 Registering persistent reservations**

To establish a persistent reservation the application client must first register with a device server. This is accomplished by issuing a PERSISTENT RESERVE OUT command with service action of REGISTER. The REGISTER service action contains a reservation key which the device server retains. The device server then allows other PERSISTENT RESERVE OUT commands from the registered initiator to execute.

Any PERSISTENT RESERVE OUT command service action, other than the REGISTER service action, received by the target results in the command being rejected with a RESERVATION CONFLICT status.

#### **0.0.1.2 Creating a persistent reservation**

An application client creates a persistent reservation held by doing the following:

- a) Issuing a PERSISTENT RESERVE OUT command with a service action of RESERVE service action through a registered initiator; and
- b) Setting the type and scope of the reservation being created.

Only one persistent reservation at a time per logical unit is allowed. If the target receives a PERSISTENT RESERVE OUT command that attempts to create a different persistent reservation than the active persistent reservation then the command shall be rejected with a RESERVATION CONFLICT status.

#### **0.0.1.3 Releasing a persistent reservation**

An application client removes an active persistent reservation held by the same initiator by doing the following:

- a) Issuing a PERSISTENT RESERVE OUT command with a service action of RELEASE service action through a registered initiator; and
- b) Setting the type and scope to match the persistent reservation being released.

In response to this request the device server shall perform a release by doing the following:

- a) Remove the selected persistent reservation; and
- b) not remove any registration key(s).

If the persistent reservation is of the type write exclusive, registrants only or exclusive access, registrants only the device server establishes a unit attention condition for all the other registered initiators. The sense key shall be set

to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. If the persistent reservation is of any other type the device server device server establishes no unit attention condition.

A RELEASE service action should not be performed if any operations interlocked by the persistent reservation are not yet complete.

#### **0.0.1.4 Replacing a persistent reservation**

An application client replaces a persistent reservation with another persistent reservation by doing the following:

- a) Issuing a PERSISTENT RESERVE OUT command with a RESERVE service action through a registered initiator;
- b) Setting the Service action reservation key to match the reservation key of the persistent reservation being replaced; and
- c) Setting the type and scope to match the persistent reservation being replaced.

In response to this request the device server replaces the current persistent reservation owner (i.e., the original initiator and reservation key) with the new persistent reservation owner (i.e., the new initiator and reservation key). The original initiator/reservation key combination shall remain registered with the device server. The only change is that the owner of the persistent reservation is now the new initiator/reservation key combination.

Any commands from any initiator that have been accepted by the device server before the new persistent reservation owner is established as nonconflicting shall continue normal execution. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a RESERVE service action when no persistent reservation exists for the initiator identified by the Service action Reservation key. In the case where no persistent reservation exists the device server shall establish the requested persistent reservation.

If the persistent reservation is of the type write exclusive, registrants only or exclusive access, registrants only the device server establishes no unit attention conditions. If the persistent reservation is of any other type the device server establishes a unit attention condition for the replaced initiator. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS REPLACED.

#### **0.0.1.5 Preempting a persistent reservation**

##### **0.0.1.5.1 PREEMPT service action**

An application client preempts a persistent reservation with another persistent reservation by doing the following:

- a) Issuing a PERSISTENT RESERVE OUT command with a PREEMPT service action through a registered initiator;
- b) Setting the Service action reservation key to match the reservation key of the persistent reservation being preempted; and
- c) Setting the type and scope of the new persistent reservation.

In response to this request the device server shall perform a preempt by doing the following:

- a) Removes the persistent reservation for the initiators identified by the Service action reservation key specified in the PERSISTENT RESERVE OUT parameter list;
- b) remove all registration key(s) except the preempting initiators (see 0.2.1.1);
- c) establish a persistent reservation for the preempting initiator; and
- d) continue normal execution of any commands from any initiator that have been accepted by the device server when the old persistent reservation was active as nonconflicting.

It shall not be an error to send a PERSISTENT RESERVE OUT specifying a PREEMPT service action when no persistent reservation exists for the initiator identified by the Service action Reservation key. In the case where no persistent reservation exists the device server shall establish the requested persistent reservation.

If the persistent reservation is of the type write exclusive, registrants only or exclusive access, registrants only the device server establishes a unit attention condition for the preempted initiator and all registered initiators. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. If the persistent reservation is of any other type the device server establishes a unit attention condition for only the preempted initiator. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED.

Subsequent commands are subject to the persistent reservation restrictions established by the preempting initiator.

#### **0.0.1.5.2 PREEMPT AND ABORT service action**

The application clients requests for and device server responses to a PERSISTENT RESERVE OUT command PREEMPT AND ABORT service action are identical to the PREEMPT service action (clause 0.0.1.5.1).

In addition the device server shall clear any ACA or CA condition associated with the initiator being preempted and shall clear any tasks with an ACA attribute from that initiator. If TST=000b (see 8.3.4), then ACA or CA conditions for initiators other than the initiator being preempted shall prevent the execution of the PERSISTENT RESERVE OUT task, which shall end with a status of ACA ACTIVE if NACA=1 (see SAM), or BUSY if NACA=0. If TST=001b, then ACA or CA conditions for initiators other than the initiator being preempted shall not prevent the execution of the PERSISTENT RESERVE OUT task.

NOTE 1 The Preempt and Abort service action will clear the ACA condition associated with the initiator being preempted even though the task is terminated with an ACA ACTIVE status. Thus, the next command arriving at the device server will not encounter the ACA condition previously active for the initiator being preempted.

---

---

Editors Note 1 - GOP: Why would the preempt and abort get an ACA ACTIVE status?

---

---

Any Asynchronous Event Reporting operations in progress that were initiated by the device server are not affected by the Preempt and Abort service action.

#### **0.0.1.6 Clearing a persistent reservation**

An application client may clear a persistent reservation and all registrations from a device server for a specific logical unit by doing the following:

- a) Issuing a PERSISTENT RESERVE OUT command with a service action of CLEAR service action through a registered initiator;

In response to this request the device server shall perform a clear by doing the following:

- a) Remove the any persistent reservation associated with the logical unit;
- b) remove all registration key(s) (see 0.2.1.1);
- c) continue normal execution of any commands from any initiator that have been accepted by the device server as nonconflicting; and
- d) establish a Unit Attention condition for all initiators for the cleared logical unit. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED.

Application clients should not use the Clear service action except during recoveries that are associated with initiator or system reconfiguration, since data integrity may be compromised.

In addition to reservations whose sharing properties depend on the SCSI domain relationships between initiators, the Persistent Reservations management method allows the creation of reservations that are shared among those initiators who register with the device server using the PERSISTENT RESERVE OUT command with the Register Service action.

## 0.1 PERSISTENT RESERVE IN command

The PERSISTENT RESERVE IN command (see table 1) is used to obtain information about persistent reservations and reservation keys that are active within a device server. This command is used in conjunction with the PERSISTENT RESERVE OUT command (see 0.2).

**Table 1 — PERSISTENT RESERVE IN command**

| Bit<br>Byte | 7                    | 6                 | 5 | 4              | 3 | 2 | 1 | 0     |
|-------------|----------------------|-------------------|---|----------------|---|---|---|-------|
| 0           | Operation code (5Eh) |                   |   |                |   |   |   |       |
| 1           | Reserved             |                   |   | Service action |   |   |   |       |
| 2           | Reserved             |                   |   |                |   |   |   |       |
| 3           | Reserved             |                   |   |                |   |   |   |       |
| 4           | Reserved             |                   |   |                |   |   |   |       |
| 5           | Reserved             |                   |   |                |   |   |   |       |
| 6           | Reserved             |                   |   |                |   |   |   |       |
| 7           | (MSB)                | Allocation length |   |                |   |   |   | (LSB) |
| 8           |                      |                   |   |                |   |   |   |       |
| 9           | Control              |                   |   |                |   |   |   |       |

When a device server receives a PERSISTENT RESERVE IN command and RESERVE(10) or RESERVE(6) logical unit or extent reservations or SMC element reservations are active (see 7.21), the command shall be rejected with a RESERVATION CONFLICT status. A PERSISTENT RESERVE IN command shall not conflict with any persistent reservation.

The actual length of the PERSISTENT RESERVE IN parameter data is available in a parameter data field. The Allocation 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 shall be returned. This shall not be considered an error. If the remainder of the list is required, the application client should send a new PERSISTENT RESERVE IN command with a Allocation length field large enough to contain the entire list.

### 0.1.1 PERSISTENT RESERVE IN Service Actions

Service actions that require access to the persistent reservation and registration information may require ~~the enabling of~~ accessing a nonvolatile memory within the logical unit. If the nonvolatile memory is not ready, the device server shall return CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense data shall be set as described in the TEST UNIT READY command (see 7.24).

The Service action codes for the PERSISTENT RESERVE IN command are defined in table 2.

**Table 2 — PERSISTENT RESERVE IN Service action codes**

| Code      | Name              | Description                                   |
|-----------|-------------------|---|
| 00h       | Read Keys         | Reads all registered Reservation Keys         |
| 01h       | Read Reservations | Reads all the current persistent reservations |
| 02h - 1Fh | Reserved          | Reserved                                      |

**0.1.1.1 Read Keys**

The Read Keys service action requests that the device server return a parameter list containing a header and a complete list of all reservation keys currently registered with the device server. If multiple initiators have registered with the same key, then that key value shall be listed multiple times, once for each such registration. The keys may have been passed by a PERSISTENT RESERVE OUT command that has performed a Register service action. The relationship between a reservation key and the initiator or port is outside the scope of this standard.

**0.1.1.2 Read Reservations**

The Read Reservations service action requests that the device server return a parameter list containing a header and a complete list of all unique the persistent reservations that are is presently active in the device server. and its extents. (Duplicate persistent reservations from the same initiator shall not be reported.)

**0.1.2 PERSISTENT RESERVE IN parameter data for Read Keys**

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the Read Keys service action is shown in table 3.

**Table 3 — PERSISTENT RESERVE IN parameter data for Read Keys**

| Bit Byte             | 7           | 6 | 5 | 4 | 3 | 2 | 1 | 0                       |  |
|----------------------|-------------|---|---|---|---|---|---|-------------------------|--|
| 0                    | (MSB) _____ |   |   |   |   |   |   | Generation              |  |
| 3                    |             |   |   |   |   |   |   | (LSB)                   |  |
| 4                    | (MSB) _____ |   |   |   |   |   |   | Additional length (n-7) |  |
| 7                    |             |   |   |   |   |   |   | (LSB)                   |  |
| Reservation key list |             |   |   |   |   |   |   |                         |  |
| 8                    | (MSB) _____ |   |   |   |   |   |   | First reservation key   |  |
| 15                   |             |   |   |   |   |   |   | (LSB)                   |  |
| .                    |             |   |   |   |   |   |   |                         |  |
| .                    |             |   |   |   |   |   |   |                         |  |
| .                    |             |   |   |   |   |   |   |                         |  |
| n-7                  | (MSB) _____ |   |   |   |   |   |   | Last reservation key    |  |
| n                    |             |   |   |   |   |   |   | (LSB)                   |  |

The Generation value is a 32-bit counter in the device server that shall be incremented every time a PERSISTENT RESERVE OUT command requests a Register, a Clear, a Preempt, or a Preempt and Abort service action operation. The counter shall not be incremented by a PERSISTENT RESERVE IN command, by a PERSISTENT RESERVE OUT command that performs a Reserve or Release service action, or by a PERSISTENT RESERVE

OUT command that is not performed due to an error or reservation conflict. Regardless of the APTPL value the Generation value shall be set to 0 as part of the power on reset process.

The Generation value allows the application client examining the generation value to verify that the configuration of the initiators attached to a logical unit has not been modified by another application client without the knowledge of the examining application client.

---



---

Editors Note 2 - GOP: The above paragraph should be in a model.

---



---

The Additional length field contains a count of the number of bytes in the Reservation key list. If the Allocation length specified by the PERSISTENT RESERVE IN command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed Allocation length shall be sent to the application client. The remaining bytes shall be truncated, although the Additional length field shall still contain the actual number of bytes in the reservation key list without consideration of any truncation resulting from an insufficient Allocation length. This shall not be considered an error.

The Reservation key list contains all the 8-byte reservation keys registered with the device server through PERSISTENT RESERVE OUT Reserve, Preempt, Preempt and Abort, or Register service actions. Each reservation key may be examined by the application client and correlated with a set of initiators and SCSI ports by mechanisms outside the scope of this standard. Duplicate keys are possible, if multiple initiators use the same reservation key.

### 0.1.3 PERSISTENT RESERVE IN parameter data for Read Reservations

The format for the parameter data provided in response to a PERSISTENT RESERVE IN command with the Read Reservations service action is shown in table 4.

**Table 4 — PERSISTENT RESERVE IN parameter data for Read Reservations**

| Bit Byte | 7        | 6 | 5 | 4 | 3    | 2 | 1 | 0                               |  |
|----------|----------|---|---|---|------|---|---|---------------------------------|--|
| 0        | (MSB)    |   |   |   |      |   |   | Generation                      |  |
| 3        |          |   |   |   |      |   |   | (LSB)                           |  |
| 4        | (MSB)    |   |   |   |      |   |   | Additional length (0 or 16-n-7) |  |
| 7        |          |   |   |   |      |   |   | (LSB)                           |  |
| 8        | (MSB)    |   |   |   |      |   |   | Reservation key                 |  |
| 15       |          |   |   |   |      |   |   | (LSB)                           |  |
| 16       | (MSB)    |   |   |   |      |   |   | Scope-specific address          |  |
| 19       |          |   |   |   |      |   |   | (LSB)                           |  |
| 20       | Reserved |   |   |   |      |   |   |                                 |  |
| 21       | Scope    |   |   |   | Type |   |   |                                 |  |
| 22       | Reserved |   |   |   |      |   |   |                                 |  |
| 23       | Reserved |   |   |   |      |   |   |                                 |  |

The Generation field shall be as defined for the PERSISTENT RESERVE IN Read Keys parameter data (see 0.1.2).

The Additional length field contains a count of the number of bytes to follow in the parameter list of Reservation descriptors. If the Allocation length specified by the PERSISTENT RESERVE IN command is not sufficient to contain the entire parameter list, then only the bytes from 0 to the maximum allowed Allocation length shall be sent to the application client. The remaining bytes shall be truncated, although the Additional length field shall still contain the actual number of bytes of Reservation descriptors and shall not be affected by the truncation. This shall not be considered an error.

The format of a single read Reservation descriptor is defined in table 5. There shall be one read Reservation descriptor for each unique persistent reservation held on the logical unit by any initiator. (Duplicate persistent reser-

Table 5 — PERSISTENT RESERVE IN Read Reservations Descriptor

| Bit<br>Byte | 7                      | 6 | 5 | 4 | 3    | 2 | 1 | 0     |
|-------------|------------------------|---|---|---|------|---|---|-------|
| 0           | (MSB)                  |   |   |   |      |   |   |       |
| 7           | Reservation key        |   |   |   |      |   |   | (LSB) |
| 8           | (MSB)                  |   |   |   |      |   |   |       |
| 11          | Scope-specific address |   |   |   |      |   |   | (LSB) |
| 12          | Reserved               |   |   |   |      |   |   |       |
| 13          | Scope                  |   |   |   | Type |   |   |       |
| 14          | (MSB)                  |   |   |   |      |   |   |       |
| 15          | Extent length          |   |   |   |      |   |   | (LSB) |

ervations from the same initiator shall not be reported.)

For each unique If a persistent reservation is held on the logical unit, there shall be a read-reservation information descriptor presented in the list of parameter data returned by the device server in response to the PERSISTENT RESERVE IN command with a Read Reservations action. The reservation information descriptor shall contain the Reservation Key under which the persistent reservation is held. The Type and Scope of the persistent reservation as present in the PERSISTENT RESERVE OUT command that created the persistent reservation shall be returned (see 0.1.3.1 and 0.1.3.2). Duplicate Reservation descriptors are possible, if multiple initiators use the same reservation key.

Reservation key is the registered reservation key under which the reservation is held. If initiators use unique keys, then the application should be able to associate the reservation key with the initiator that holds the reservation. This association is done using techniques that are outside the scope of this standard.

---



---

Editors Note 3 - GOP: The above paragraph should be in a model.

---



---

If the Scope is an Extent reservation, the Scope-specific address field shall contain the LBA of the first block of the extent and the Extent length field shall contain the number of blocks in the extent. If the Scope is an Element reservation, the Scope-specific address field shall contain the Element address, zero filled in the most significant bytes to fit the field, and the Extent length field shall be set to zero. If the Scope is a Logical Unit reservation, both the Scope-specific address and Extent length fields shall be set to zero.

### 0.1.3.1 Persistent Reservations Scope

The value in the Scope field shall indicate 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.~~ The values in the Scope field are defined in table 6.

**Table 6 — Persistent Reservation Scope Codes**

| Code    | Name                                 | Description  |
|---------|--------------------------------------|--|
| 0h      | LU                                   | Persistent reservation applies to the full logical unit                              |
| 1h      | <del>Extent</del><br><u>Reserved</u> | <del>Persistent reservation applies to the specified extent</del><br><u>Reserved</u> |
| 2h      | Element                              | Persistent reservation applies to the specified element                              |
| 3h - Fh | Reserved                             | Reserved   |

#### 0.1.3.1.1 LU Scope

A Scope field value of LU shall indicate that the persistent reservation applies to the entire logical unit. The LU scope shall be implemented by all device servers that implement PERSISTENT RESERVE OUT.

#### 0.1.3.1.2 Extent Scope

~~A Scope field value of Extent shall indicate that the persistent reservation applies to the extent of the logical unit defined by the Scope-specific address and Extent length fields in the PERSISTENT RESERVE OUT parameter list. An extent is defined only for devices defining contiguous logical block addresses. The Extent scope is optional for all device servers that implement PERSISTENT RESERVE OUT. The number of extents that may be reserved for a logical unit is vendor-specific.~~

#### 0.1.3.1.3 Element Scope

A Scope field value of Element shall indicate that the persistent reservation applies to the element of the logical unit defined by the Scope-specific address field in the PERSISTENT RESERVE OUT parameter list. An element is defined by the SCSI-3 Medium Changer Commands (SMC) standard. The Element scope is optional for all device servers that implement PERSISTENT RESERVE OUT.

### 0.1.3.2 Persistent Reservations Type

The value in the Type field shall specify the characteristics of the persistent reservation being established for all data blocks within the extent or within the logical unit. Table 7 defines the characteristics of the ~~five~~ different type values. For each persistent reservation type, table 7 lists code value and describes the required device server support. In table 7, the description of required device server support is divided into ~~two three~~ paragraphs. The first paragraph defines the required handling for read operations. The second paragraph defines the required handling for write operations. ~~The third paragraph defines the handling for subsequent attempts to establish persistent reservations.~~



Table 7 — Persistent Reservation Type Codes (part 1 of 3)

| Code | Name                 | Description  |
|------|----------------------|--|
| 0h   | Read Shared Reserved | <p><del>Reads Shared: Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator.</del></p> <p><del>Writes Prohibited: Any command from any initiator that performs a transfer from the initiator to the storage medium or cache of the logical unit shall result in a reservation conflict.</del></p> <p><del>Additional Reservations Allowed: Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</del></p> <hr/> <p>Editors Note 4 - GOP: Of no value. Defines the case where any initiator can read but no initiator can write. This is true regardless of which initiator holds the reservation.</p>  |
| 1h   | Write Exclusive      | <p><del>Reads Shared: Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator.</del></p> <p><b>Writes Exclusive:</b> Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the initiator to the storage medium or cache of the logical unit shall result in a reservation conflict.</p> <p><del>Additional Reservations Allowed: Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</del></p>   |
| 2h   | Read Exclusive       | <p><b>Reads Exclusive:</b> Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the storage medium or cache of the logical unit to the initiator shall result in a reservation conflict.</p> <p><b>Writes Shared:</b> Any application client on any initiator may execute commands that perform transfers from the initiator to the storage medium or cache of the logical unit.</p> <p><del>Additional Reservations Allowed: Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</del></p> <hr/> <p>Editors Note 5 - GOP: Of marginal value. I would like it removed unless someone can come up with a reasonable use for this option. Defines the case where only the initiator holding the reservation can read but any initiator can write.</p> |

Table 7 — Persistent Reservation Type Codes (part 2 of 3)

| Code | Name                              | Description  |
|------|-----------------------------------|--|
| 3h   | Exclusive Access                  | <p><b>Reads Exclusive:</b> Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the storage medium or cache of the logical unit to the initiator shall result in a reservation conflict.</p> <p><b>Writes Exclusive:</b> Any command from any initiator other than the initiator holding the persistent reservation that performs a transfer from the initiator to the storage medium or cache of the logical unit shall result in a reservation conflict.</p> <p><b>Additional Reservations Restricted:</b> Any Persistent Reserve Out command with the Reserve service action from any initiator other than the initiator holding the persistent reservation shall result in a reservation conflict. The initiator that holds the persistent reservation may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</p>                                |
| 4h   | Shared Access Reserved            | <p><b>Reads Shared:</b> Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator.</p> <p><b>Writes Shared:</b> Any application client on any initiator may execute commands that perform transfers from the initiator to the storage medium or cache of the logical unit.</p> <p><b>Additional Reservations Restricted:</b> Any Persistent Reserve Out command with the Reserve service action from any initiator other than the initiator holding the persistent reservation shall result in a reservation conflict. The initiator that holds the persistent reservation may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</p> <hr/> <p>Editors Note 6 - GOP: This is of no value. It defines the case where there is no reservation. (i.e. any initiator can read or write regardless of reservation).</p> <hr/> |
| 5h   | Write Exclusive, Registrants Only | <p><b>Reads Shared:</b> Any application client on any initiator may execute commands that perform transfers from the storage medium or cache of the logical unit to the initiator.</p> <p><b>Writes Exclusive:</b> Any command from an initiator that has not previously performed a Register service action with the device server that performs a transfer to the storage medium or cache of the logical unit, shall result in a reservation conflict.</p> <p><b>Additional Reservations Allowed:</b> Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</p>   |

Table 7 — Persistent Reservation Type Codes (part 3 of 3)

| Code    | Name                               | Description  |
|---------|------------------------------------|--|
| 6h      | Exclusive Access, Registrants Only | <p><b>Reads Exclusive:</b> Any command from an initiator that has not previously performed a Register service action with the device server that performs a transfer from the storage medium or cache of the logical unit, shall result in a reservation conflict.</p> <p><b>Writes Exclusive:</b> Any command from an initiator that has not previously performed a Register service action with the device server that performs a transfer to the storage medium or cache of the logical unit, shall result in a reservation conflict.</p> <p><b>Additional Reservations Allowed:</b> Any initiator may reserve the logical unit or extents or elements as long as the persistent reservations do not conflict with any reservations that are already known to the device server. See table 8.</p> |
| 7h - Fh | Reserved                           |  |

Table 8 — New Persistent Reservation Conflicts With Existing

| Persistent Reservation That Is Being Attempted | Persistent Reservation That is Held |    |                 |    |                |    |                    |    |                 |    |                    |    |                     |    |   |
|--|-------------------------------------|----|-----------------|----|----------------|----|--------------------|----|-----------------|----|--------------------|----|---------------------|----|---|
|  | Read-Shared                         |    | Write-Exclusive |    | Read-Exclusive |    | Exclusive Access † |    | Shared Access † |    | Write-Exclusive RO |    | Exclusive Access-RO |    |   |
|  | LU                                  | EX | LU              | EX | LU             | EX | LU                 | EX | LU              | EX | LU                 | EX | LU                  | EX |   |
| Read-Shared                                    | LU                                  | N  | N               | Y  | Y              | Y  | Y                  | Y  | Y               | N  | N                  | Y  | Y                   | Y  | Y |
|  | EX                                  | N  | N               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | N  | N                  | Y  | Ø                   | Y  | Ø |
| Write-Exclusive                                | LU                                  | Y  | Y               | Y  | Y              | Y  | Y                  | Y  | Y               | Y  | Y                  | Y  | Y                   | Y  | Y |
|  | EX                                  | Y  | Ø               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | Y  | Ø                  | Y  | Ø                   | Y  | Ø |
| Read-Exclusive                                 | LU                                  | Y  | Y               | Y  | Y              | Y  | Y                  | Y  | Y               | Y  | Y                  | Y  | Y                   | Y  | Y |
|  | EX                                  | Y  | Ø               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | Y  | Ø                  | Y  | Ø                   | Y  | Ø |
| Exclusive Access †                             | LU                                  | Y  | Y               | Y  | Y              | Y  | Y                  | Y  | Y               | Y  | Y                  | Y  | Y                   | Y  | Y |
|  | EX                                  | Y  | Ø               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | Y  | Ø                  | Y  | Ø                   | Y  | Ø |
| Shared Access †                                | LU                                  | N  | N               | Y  | Y              | Y  | Y                  | Y  | Y               | N  | N                  | Y  | Y                   | Y  | Y |
|  | EX                                  | N  | N               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | N  | N                  | Y  | Ø                   | Y  | Ø |
| Write-Exclusive RO                             | LU                                  | Y  | Y               | Y  | Y              | Y  | Y                  | Y  | Y               | Y  | Y                  | N  | N                   | N  | N |
|  | EX                                  | Y  | Ø               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | Y  | Ø                  | N  | N                   | N  | N |
| Exclusive Access-RO                            | LU                                  | Y  | Y               | Y  | Y              | Y  | Y                  | Y  | Y               | Y  | Y                  | N  | N                   | N  | N |
|  | EX                                  | Y  | Ø               | Y  | Ø              | Y  | Ø                  | Y  | Ø               | Y  | Ø                  | N  | N                   | N  | N |

Key:

- LU = Logical Unit scope
- EX = Extent or Element scope
- RO = Registrants Only
- † = Conflicts with all reservation requests from other initiators
- N = no conflict
- Y = conflict
- Ø = conflict occurs if extent or element overlaps with existing extent or element reservation

## 0.2 PERSISTENT RESERVE OUT command

The PERSISTENT RESERVE OUT command (see table 9) 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 shall be used in conjunction with the PERSISTENT RESERVE IN command and shall not be used with the RESERVE and RELEASE commands.

Persistent reservations shall conflict with reservations established by the RESERVE command. Initiators performing PERSISTENT RESERVE OUT Service actions are identified by a reservation key provided by the application client. An application client may use the PERSISTENT RESERVE IN command to identify which initiators ~~are is holding conflicting or invalid a~~ persistent reservations and use the PERSISTENT RESERVE OUT command to preempt ~~those~~ that reservations if required.

Only one persistent reservation shall be in effect per logical unit or element. Any attempt to create another persistent reservation using a persistent reservation out command shall return a RESERVATION CONFLICT status except as noted below.

Since persistent reservations are not reset by the TARGET RESET task management function or other global actions, they may be used to enforce device sharing among multiple initiators. The PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands provide the basic mechanism for dynamic contention resolution in multiple initiator systems using multiple port targets. The identification of persistent reservations using the reservation key makes it possible to determine which ports hold conflicting persistent reservations and to take over persistent reservations from failing or uncooperative initiators.

---



---

Editors Note 7 - GOP: The above paragraph should be in a model.

---



---

**Table 9 — PERSISTENT RESERVE OUT command**

| Bit Byte | 7                    | 6                           | 5 | 4              | 3    | 2 | 1 | 0     |  |
|----------|----------------------|-----------------------------|---|----------------|------|---|---|-------|--|
| 0        | Operation code (5Fh) |                             |   |                |      |   |   |       |  |
| 1        | Reserved             |                             |   | Service action |      |   |   |       |  |
| 2        | Scope                |                             |   |                | Type |   |   |       |  |
| 3        | Reserved             |                             |   |                |      |   |   |       |  |
| 4        | Reserved             |                             |   |                |      |   |   |       |  |
| 5        | Reserved             |                             |   |                |      |   |   |       |  |
| 6        | Reserved             |                             |   |                |      |   |   |       |  |
| 7        | (MSB)                | Parameter list length (18h) |   |                |      |   |   |       |  |
| 8        |                      |                             |   |                |      |   |   | (LSB) |  |
| 9        | Control              |                             |   |                |      |   |   |       |  |

When a device server receives a PERSISTENT RESERVE OUT command and RESERVE(10) or RESERVE(6) logical unit ~~or extent reservations~~ or SMC element reservations are active (see 7.21), the command shall be rejected with a RESERVATION CONFLICT status.

A PERSISTENT RESERVE OUT command with any service action except Register shall be rejected with RESERVATION CONFLICT status if the reserving initiator does not have a registered reservation key.

If a PERSISTENT RESERVE OUT command 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 data shall be set to INSUFFICIENT RESERVATION RESOURCES.

The PERSISTENT RESERVE OUT command contains fields that specify a persistent reservation Service action, the intended scope of the persistent reservation, and the restrictions caused by the persistent reservation. The Type and Scope fields are defined in 0.1.3.1 and 0.1.3.2. If a Scope field specifies a scope that is not implemented, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID FIELD IN CDB.

Fields contained in the PERSISTENT RESERVE OUT parameter list specify the reservation keys and extent information required to perform a particular persistent reservation Service action.

The parameter list shall be 24 bytes in length and the Parameter list length field shall contain 24 (18h). If the Parameter list length is not 24, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and the additional sense data shall be set to PARAMETER LIST LENGTH ERROR.

The capability of preserving persistent reservations and registration keys requires the use of a nonvolatile memory within the logical unit. If the nonvolatile memory is not accessible at the time that a PERSISTENT RESERVE OUT command attempts to activate the Persist Through Power Loss capability, the device server shall return CHECK CONDITION status. The sense key shall be set to NOT READY and the additional sense data shall be set as described in the TEST UNIT READY command (see 7.24).

**0.2.1 PERSISTENT RESERVE OUT Service Actions**

When processing the PERSISTENT RESERVE OUT service actions, the device server shall increment the generation value as specified in 0.1.2.

The PERSISTENT RESERVE OUT command Service actions are defined in table 10.

**Table 10 — PERSISTENT RESERVE OUT Service action codes**

| Code      | Name            | Description   |
|-----------|-----------------|---|
| 00h       | Register        | Register a reservation key with the device server   |
| 01h       | Reserve         | Create a persistent reservation using a reservation key   |
| 02h       | Release         | Release a persistent reservation  |
| 03h       | Clear           | Clear all reservation keys and all persistent reservations  |
| 04h       | Preempt         | Preempt persistent reservations from another initiator  |
| 05h       | Preempt & abort | Preempt persistent reservations from another initiator and abort the task set for the preempted initiator |
| 06h - 1Fh | Reserved        |   |

**0.2.1.1 Register**

A PERSISTENT RESERVE OUT command with a service action of Register shall not conflict with any persistent reservation by any initiator.

The PERSISTENT RESERVE OUT command executing a Register service action registers a reservation key with a device server without generating a reservation. 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:

- a) powering down the logical unit, if the last APTPL received by the device server was zero (see 0.2.2);
- b) performing a Clear service action;
- c) performing a Preempt service action;
- d) performing a Preempt and Abort service action; or
- e) performing a Register service action from the same initiator with the value of the Service action Reservation key field set to zero.

When a reservation key has not yet been established or when the reservation key has been removed, a reservation key of zero shall be used when the initiator performs a PERSISTENT RESERVE OUT with the Register service action. When the reservation has been removed, no information shall be reported for the initiator in the Report Keys service action.

The Register service action shall be processed regardless of any active persistent reservations. ~~Any~~ All existing persistent reservations for the initiator shall be updated to reflect the new reservation key. The device server shall ignore the contents of the Scope and Type fields.

#### 0.2.1.2 Reserve

A PERSISTENT RESERVE OUT command with a service action of Reserve shall be rejected with RESERVATION CONFLICT status and the reservation shall not be made if the new reservation does not contain the same reservation key in the Service Action Reservation key field (see table 11) as the existing reservation or conflicts in scope or type, or extent with any persistent reservation by any initiator at the time the PERSISTENT RESERVE OUT with Reserve service action is enabled for execution.

The PERSISTENT RESERVE OUT command performing a Reserve service action creates a persistent reservation having a specified scope and type. The scope and type of a persistent reservation are defined in 0.1.3.1 and 0.1.3.2.

Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt service action, or Preempt and Abort service action or a Reserve service action having a type of read shared, shared access, write exclusive registrants only, and exclusive access registrants only if the type of the existing reservation and the requested reservation are the same. The device server shall replace the existing persistent reservation with the requested persistent reservation. A unit attention condition is established for the preempted initiator. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS REPLACED. 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 shall all be simultaneously released by a single Release service action that matches the reservations.

---



---

Editors Note 8 - GOP: RESERVATIONS REPLACED is a new ASC.

---



---

A persistent reservation shall ~~be tested for conflicts with other persistent reservations as described above and shall~~ take effect when the task executing the PERSISTENT RESERVE OUT command enters the enabled task state.

### 0.2.1.3 Release

A PERSISTENT RESERVE OUT command with a service action of Release shall not conflict with any persistent reservation by any initiator.

The PERSISTENT RESERVE OUT command performing a Release service action removes an active persistent reservation held by the same initiator. The fields associated with the Release service action shall match fields of the active persistent reservation. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Release service action when no persistent reservation exists from that initiator. In this case, the device server shall return GOOD status without altering any other reservation. The reservation key shall not be changed by the Release service action.

---

---

Editors Note 9 - GOP: The above paragraph should be moved to the model.

---

---

The device server shall return a CHECK CONDITION status for a PERSISTENT RESERVE OUT command that specifies the release of a persistent reservation held by the requesting initiator matching some but not all of the scope, type, reservation key, and extent values. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID RELEASE OF ACTIVE PERSISTENT RESERVATION. ~~Attempts to release persistent reservations where none of the scope, type, reservation key, and extent values match an existing persistent reservation held by the requesting initiator shall not be considered errors.~~

---

---

Editors Note 10 - GOP: What is the point of the strikethrough text above? This looks like a no-op why have this special condition?

---

---

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

- a) Power off. When the most recent APTPL value received by the device server is zero (see 0.2.2), a power off performs a hard reset, clears all persistent reservations, and removes all registered reservation keys (see 0.2.1.1); or
- b) Execution of a PERSISTENT RESERVE OUT command from another initiator with a Persistent Reserve service action of Preempt or Preempt and Abort.

A Release service action should not be performed if any operations interlocked by the persistent reservation are not yet complete.

---

---

Editors Note 11 - GOP: The above paragraph should be moved to the model.

---

---

### 0.2.1.4 Clear

A PERSISTENT RESERVE OUT command with a service action of Clear shall not conflict with any persistent reservation by any initiator.

A PERSISTENT RESERVE OUT command with a Clear service action from a registered initiator shall cause the removal of ~~The PERSISTENT RESERVE OUT command that successfully performs a Clear service action shall remove~~ all persistent reservations for all initiators. All reservation keys shall be removed (see 0.2.1.1). Any commands from any initiator that have been accepted by the device server as nonconflicting shall continue normal execution.

A Unit Attention condition shall be established for all initiators for the cleared logical unit. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED.

Application clients should not use the Clear service action except during recoveries that are associated with initiator or system reconfiguration, since data integrity may be compromised.

---



---

Editors Note 12 - GOP: The above three paragraphs should go to the model.

---



---

The device server shall ignore the contents of the Scope and Type fields.

#### 0.2.1.5 Preempt

A PERSISTENT RESERVE OUT command with a service action of Preempt shall not conflict with any persistent reservation for the initiator(s) being preempted. If the Service Action Reservation key (see table 11) does not match any registered reservation key the device server shall return a RESERVATION CONFLICT status. ~~If there is a reservation by any initiator (except the initiator(s) being preempted) that conflicts with the new reservation being established by the initiator executing the PERSISTENT RESERVE OUT command with Preempt service action, the PERSISTENT RESERVE OUT command shall be rejected with RESERVATION CONFLICT status and the Preempt service action shall not be performed.~~

The PERSISTENT RESERVE OUT command that successfully performs a Preempt service action shall;

- a) remove the all persistent reservations for the all initiators ~~that are registered~~ identified by with the Service action Reservation key specified in the PERSISTENT RESERVE OUT parameter list;
- b) remove all registration key(s) for the other initiators that have been preempted (see 0.2.1.1); and
- c) ~~It shall also establish a persistent reservation for the preempting initiator.~~

Any commands from any initiator that have been accepted by the device server when the old persistent reservation was active as nonconflicting shall continue normal execution. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Preempt service action when no persistent reservation exists for the initiator identified by the Service action Reservation key.

A Unit Attention condition is established for the preempted initiators and all registered initiators if the preempted reservation is a write exclusive, registrants only or a exclusive access, registrants only. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. Subsequent commands are subject to the persistent reservation restrictions established by the preempting initiator.

---



---

Editors Note 13 - GOP: The above three paragraph should be moved to the model.

---



---

The persistent reservation created by the preempting initiator is specified by the scope and type field of the PERSISTENT RESERVE OUT command and the corresponding fields in the PERSISTENT RESERVE OUT parameter list. The scope and type of the persistent reservation created by preempting initiator may be different than the persistent reservation being preempted.

~~The registration key for the other initiators that have been preempted shall be removed (see 0.2.1.1) by the Preempt service action. The reservation key for an initiator that has performed a Preempt service action with its own Reservation key specified in the Service action Reservation key shall remain unchanged, although all other specified releasing actions and reservation actions shall be performed.~~



~~Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt or Preempt and Abort service action. 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 other than the reservations being preempted are held by another initiator.~~

### 0.2.1.6 Preempt and Abort

The PERSISTENT RESERVE OUT command performing a Preempt and Abort service action, in addition to the requirements in this clause, have the same requirements as defined for the Preempt service action (see clause 0.2.1.6).

~~A PERSISTENT RESERVE OUT command with a service action of Preempt and Abort shall not conflict with any persistent reservation for the initiator(s) being preempted and aborted. If the Service Action Reservation key (see table 11) does not match any registered reservation key the device server shall return a RESERVATION CONFLICT status. If there is a reservation by any initiator (except the initiator(s) being preempted and cleared) that conflicts with the new reservation being established by the initiator executing the PERSISTENT RESERVE OUT command with Preempt and Abort service action, the PERSISTENT RESERVE OUT command shall be rejected with RESERVATION CONFLICT status and the Preempt and Abort service action shall not be performed.~~

~~The PERSISTENT RESERVE OUT command performing a Preempt and Abort service action removes all persistent reservations for all initiators that are registered with the Service Action Reservation key specified by the PERSISTENT RESERVE OUT parameter list. It shall also establish a persistent reservation for the preempting initiator. Every command from the initiators being preempted shall be terminated as if an ABORT TASK task management function had been performed by the preempted initiator. It shall not be an error to send a PERSISTENT RESERVE OUT specifying a Preempt and Abort service action when no persistent reservation exists for the initiator identified by the Service Action Reservation key. However, if the key is registered, the Clear portion of the action shall execute normally.~~

~~A Unit Attention condition is established for the preempted initiators. The sense key shall be set to UNIT ATTENTION and the additional sense data shall be set to RESERVATIONS PREEMPTED. Subsequent new commands and retries of commands that timed out because they were cleared are subject to the persistent 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 PERSISTENT RESERVE OUT command and the corresponding fields in the PERSISTENT RESERVE OUT parameter list.~~

~~The Preempt and Abort service action shall clear any ACA or CA condition associated with the initiator being preempted and shall clear any tasks with an ACA attribute from that initiator. If TST=000b (see 8.3.4), then ACA or CA conditions for other initiators other than the initiator being preempted shall prevent the execution of the PERSISTENT RESERVE OUT task, which shall end with a status of ACA ACTIVE if NACA=1 (see SAM), or BUSY if NACA=0. If TST=001b, then ACA or CA conditions for other initiators other than the initiator being preempted shall not prevent the execution of the PERSISTENT RESERVE OUT task.~~

NOTE 2 The Preempt and Abort service action will clear the ACA condition associated with the initiator being preempted even though the task is terminated with an ACA ACTIVE status. Thus, the next command arriving at the device server will not encounter the ACA condition previously active for the initiator being preempted.

---

---

Editors Note 14 - GOP: Why would the preempt and abort get an ACA ACTIVE status?

---

---

Any Asynchronous Event Reporting operations in progress that were initiated by the device server are not affected by the Preempt and Abort service action.

---



---

Editors Note 15 - GOP: The above three paragraphs should be moved to the model.

---



---

The reservation key for the other initiators that have been preempted shall be removed (see 0.2.1.1) by the Preempt and Abort service action. The reservation key for an initiator that has performed a Preempt and Abort service action with its own Reservation key specified in the Service action Reservation key shall remain unchanged, although all other specified clearing actions, releasing actions and reservation actions shall be performed.

Persistent reservations shall not be superseded by a new persistent reservation from any initiator except by execution of a PERSISTENT RESERVE OUT specifying either the Preempt or Preempt and Abort service action. 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 other than the reservations being preempted are held by another initiator.

## 0.2.2 PERSISTENT RESERVE OUT parameter list

The parameter list required to perform the PERSISTENT RESERVE OUT command are defined in table 11. All fields shall be sent on all PERSISTENT RESERVE OUT commands, even if the field is not required for the specified Service action and Scope values.

**Table 11 — PERSISTENT RESERVE OUT parameter list**

| Bit<br>Byte | 7     | 6                              | 5 | 4 | 3 | 2 | 1 | 0     |
|-------------|-------|--------------------------------|---|---|---|---|---|-------|
| 0           | (MSB) | Reservation key                |   |   |   |   |   | (LSB) |
| 7           |       |                                |   |   |   |   |   |       |
| 8           | (MSB) | Service action Reservation key |   |   |   |   |   | (LSB) |
| 15          |       |                                |   |   |   |   |   |       |
| 16          | (MSB) | Scope-specific address         |   |   |   |   |   | (LSB) |
| 19          |       |                                |   |   |   |   |   |       |
| 20          |       | Reserved                       |   |   |   |   |   | APTPL |
| 21          |       | Reserved                       |   |   |   |   |   |       |
| 22          | (MSB) | Extent length                  |   |   |   |   |   | (LSB) |
| 23          |       |                                |   |   |   |   |   |       |

The Reservation key field contains an 8-byte ~~token value~~ provided by the application client to the device server to identify the initiator that is the source of the PERSISTENT RESERVE OUT command. The device server shall verify that the Reservation key field in a PERSISTENT RESERVE OUT command matches the registered reservation key for the initiator from which the command was received except for the Register service action for an unregistered initiator which shall have a reservation key value of zero. If a PERSISTENT RESERVE OUT command specifies a Reservation key field other than the reservation key registered for the initiator, the device server shall return a RESERVATION CONFLICT status. The reservation key of the initiator shall be valid for all Service action and Scope values.

The Service Action Reservation key field contains information needed for ~~three~~ the following service actions; the Reserve, Register, Preempt, and Preempt and Abort service actions. For the Reserve service action, the Service Action Reservation key field optionally contains the reservation key of the persistent reservation being replaced. For the Register service action, the Service Action Reservation key field contains the new reservation key to be registered. For the Preempt and Preempt and Abort service actions, the Service Action Reservation key field contains the reservation key of the persistent reservations that are being preempted. For the Preempt and Preempt and Abort service actions, failure of the Service Action Reservation key to match any registered reservation keys shall result in the device server returning a RESERVATION CONFLICT status. The Service Action Reservation key is ignored for all service actions except those described in this paragraph.

~~If the Scope is an Extent reservation, the Scope-specific address field shall contain the LBA of the first block of the extent and the Extent length field shall contain the number of blocks in the extent. If the Scope is an Element reservation, the Scope-specific address field shall contain the Element address, zero filled in the most significant bytes to fit the field, and the Extent length field shall be set to zero. If the Service action is Register or Clear or if the Scope is a Logical Unit reservation, both the Scope-specific address and Extent length fields shall be set to zero.~~

The Activate Persist Through Power Loss (APTPL) bit shall be valid only for the Register service action. In all other cases, the APTPL shall be ignored. Support for an APTPL bit equal to one is optional. If a device server that does not support the APTPL bit value of one receives that value in a Register service action, the device server shall return a CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST and additional sense data shall be set to INVALID FIELD IN PARAMETER LIST.

If the last valid APTPL bit value received by the device server is zero, the loss of power in the target shall release the all-persistent reservations for all logical units and remove all reservation keys (see 0.2.1.1). If the last valid APTPL bit value received by the device server is one, the logical unit shall retain all the persistent reservations for all logical units and all reservation keys for all initiators even if power is lost and later returned. The most recently received valid APTPL value from any initiator shall govern logical unit's behavior in the event of power loss.

Table 46 summarizes which fields are set by the application client and interpreted by the device server for each Service action and Scope value. Two PERSISTENT RESERVE OUT parameters are not summarized in table 46; Reservation key and APTPL, since they are specified above.

**Table 12 — PERSISTENT RESERVE OUT Service actions and valid parameters**

| Service action | Allowed Scope | Parameters |                                |                               |
|----------------|---------------|------------|--------------------------------|-------------------------------|
|                |               | Type       | Service Action Reservation key | Element or Element Parameters |
| Register       | ignored       | ignored    | valid                          | ignored                       |
| Reserve        | LU            | valid      | ignored                        | ignored                       |
| Reserve        | Extent        | valid      | ignored                        | Extent valid                  |
| Reserve        | Element       | valid      | ignored                        | Element valid                 |
| Release        | LU            | valid      | ignored                        | ignored                       |
| Release        | Extent        | valid      | ignored                        | Extent valid                  |
| Release        | Element       | valid      | ignored                        | Element valid                 |
| Clear          | ignored       | ignored    | ignored                        | ignored                       |

**Table 12 — PERSISTENT RESERVE OUT Service actions and valid parameters**

| Service action   | Allowed Scope                      | Parameters                         |                                    |   |
|--|------------------------------------|------------------------------------|------------------------------------|---|
|  |                                    | Type                               | Service Action Reservation key     | Element or Element Parameters                       |
| Preempt<br><del>Preempt</del><br>Preempt                             | LU<br><del>Extent</del><br>Element | valid<br><del>valid</del><br>valid | valid<br><del>valid</del><br>valid | ignored<br><del>Extent valid</del><br>Element valid |
| Preempt & abort<br><del>Preempt &amp; abort</del><br>Preempt & abort | LU<br><del>Extent</del><br>Element | valid<br><del>valid</del><br>valid | valid<br><del>valid</del><br>valid | ignored<br><del>Extent valid</del><br>Element valid |