

Document: T10/00-219r0

Date: July 13, 2001

To: T10 Committee Membership

From: Edward A. Gardner, Ophidian Designs

Subject: Interlocking BUSY, RESERVATION CONFLICT and TASK SET FULL

---

The various serial protocols (FCP, SRP, iSCSI) share a possible problem regarding BUSY, RESERVATION CONFLICT and TASK SET FULL. If a system environment where both command queuing and command ordering are important, there is at present no satisfactory interlock between a command being rejected with one of these status codes and subsequent commands. This is particularly a concern for iSCSI, which contemplates operating in environments with larger communication latencies than traditional SCSI environments.

Several CAP working group meetings ago we agreed to resolve this issue by reporting these conditions with CHECK CONDITION and a suitable ASC/ASCQ rather than separate status codes as at present. A bit in the control mode page would determine which behavior is implemented. This proposal documents the changes to SAM-2 and SPC-3 to accomplish this.

The general approach is to define "busy condition", etc. in SAM-2. Add a sub-clause to SAM-2 defining a busy condition and stating that it shall be reported either with BUSY status or as a CHECK CONDITION with a specified sense key and additional sense code, depending on the setting of the control mode page bit. Then replace every occurrence of "return BUSY status" with "report a busy condition". This proposal addresses every reference to the three status codes in SAM-2 and SPC-3; it does not address any reference to them in other standards.

I believe the proper sense key to use is ABORTED COMMAND:

ABORTED COMMAND. Indicates that the device server aborted the command. The application client may be able to recover by trying the command again.

SAM-2 presently specifies that this sense key be used when reporting an overlapped command (duplicate task tag detected). SPC-3 presently specifies that this sense key be used for a READ BUFFER command when the echo buffer has been overwritten.

I am naming the additional sense codes BUSY CONDITION REPORTED, etc. Specific ASC/ASCQ code value assignments are left to the SPC-3 editor.

I should state that I do not entirely agree with this proposal. I am concerned about the additional complexity for bridge controllers having to track changes to the control mode page. I would prefer that the bit controlling this behavior be in the control byte of every CDB rather than a mode page. I'm also concerned that those wishing to use this feature may not have thought through the implications of a reservation conflict creating an ACA condition, resulting in possible denial of service to the reservation holder.

Changes to SAM-2 (sam2r18.pdf).

Sub-clause 3.1, page 6, pdf page 28, add the following definitions:

busy condition: A condition indicating that the logical unit is unable to accept a command from an otherwise acceptable initiator (see 5.x.1).

rejected command condition: A condition indicating that the logical unit completed a command before entering it into a task set.

reservation conflict condition: A condition indicating that an initiator attempted to access a logical unit or an element of a logical unit that is reserved with a conflicting reservation type for another initiator (see 5.x.2).

task set full condition: A condition indicating that a lack of task set resources prevents entering a newly received tagged task into the task set (see 5.x.3).

Sub-clause 5.3.1, pages 52-53, pdf pages 74-75, replace the definitions of BUSY, RESERVATION CONFLICT and TASK SET FULL with the following:

BUSY. This status may be returned to report a busy condition (see 5.x.1).

RESERVATION CONFLICT. This status may be returned to report a reservation conflict condition (see 5.x.2).

TASK SET FULL. This status may be returned to report a task set full condition (5.x.3).

Clause 5, add the following new subclauses:

5.x Rejected command conditions

5.x.1 Busy condition

A busy condition indicates that the logical unit is busy. This condition shall be reported whenever a logical unit is unable to accept a command from an otherwise acceptable initiator (i.e., no reservation conflicts). The recommended initiator recovery action is to issue the command again at a later time.

If the rejected command interlock bit (RCINTLCK) of the logical unit's control mode page (see SPC-3) contains zero, the logical unit shall report a busy condition by returning BUSY status.

If the rejected command interlock bit (RCINTLCK) of the logical unit's control mode page (see SPC-3) contains one, the logical unit shall report a busy condition by returning CHECK CONDITION status. The sense key shall be set to ABORTED COMMAND and the additional sense code shall be set to BUSY CONDITION REPORTED.

5.x.2 Reservation conflict condition

A reservation conflict condition shall be reported whenever an initiator attempts to access a logical unit or an element of a logical unit that is reserved with a conflicting reservation type for another SCSI initiator. (See the RESERVE, RELEASE, PERSISTENT RESERVE OUT and PERSISTENT RESERVE IN commands in the SPC-2 standard). The recommended initiator recovery action is to issue the command again at a later time. Removing a persistent reservation belonging to a failing initiator may require the processing of a PERSISTENT RESERVE OUT command with the Preempt or Preempt and Clear actions (see the SPC-2 standard).

If the rejected command interlock bit (RCINTLCK) of the logical unit's control mode page (see SPC-3) contains zero, the logical unit shall report a reservation conflict condition by returning RESERVATION CONFLICT status.

If the rejected command interlock bit (RCINTLCK) of the logical unit's control mode page (see SPC-3) contains one, the logical unit shall report a reservation conflict condition by returning CHECK CONDITION status. The sense key shall be set to ABORTED COMMAND and the additional sense code shall be set to RESERVATION CONFLICT CONDITION REPORTED.

5.x.3 Task set full condition

This condition shall be implemented if the logical unit supports the creation of tagged tasks (see 4.9). This condition shall not be implemented if the logical unit does not support the creation of tagged tasks.

When the logical unit has at least one task in the task set for an initiator and a lack of task set resources prevents entering a newly received tagged task from that initiator in the task set, a task set full condition shall be reported. When the logical unit has no task in the task set for an initiator and a lack of task set resources prevents entering a newly received tagged task from that initiator in the task set, a busy condition should be reported.

When the logical unit has at least one task in the task set and a lack of task set resources prevents entering a newly received untagged task in the task, a busy condition should be reported.

The logical unit should allow at least one queued command for each supported initiator that has identified itself to the target by a protocol specific procedure or by the successful transmission of a command.

If the rejected command interlock bit (RCINTLCK) of the logical unit's control mode page (see SPC-3) contains zero, the logical unit shall report a task set full condition by returning TASK SET FULL status.

If the rejected command interlock bit (RCINTLCK) of the logical unit's control mode page (see SPC-3) contains one, the logical unit shall report a task set full condition by returning CHECK CONDITION status. The sense key shall be set to ABORTED COMMAND and the additional sense code shall be set to TASK SET FULL CONDITION REPORTED.

[Note: the sub-clause discussing overlapped commands might be moved here, as that is another reason for rejecting a command.]

Sub-clause 5.8.5, Unit Attention condition, top of page 66, pdf page 88, fifth paragraph. Reference to BUSY status removed by 00-359r2.

Sub-clause 7.1, Introduction to task set management, page 74, pdf page 96, third paragraph:

The rules for task set management only apply to a task after it has been entered into a task set. A task shall be entered into a task set unless a condition exists that causes that task to be completed with a status of BUSY, RESERVATION CONFLICT, TASK SET FULL, ACA ACTIVE or CHECK CONDITION (if **reporting** an overlapped command **or a rejected command condition, see 5.8.2 and 5.x**). A task may also be completed because of a CHECK CONDITION status caused by certain protocol specific errors.

Changes to SPC-3 (spc3r00.pdf):

Sub-clause 5.5.1, Reservations overview, third paragraph from top of page 24, pdf page 43:

The two methods are prevented from creating conflicting and undefined interactions using **reservation conflict conditions** in the following manner. If a logical unit has executed a PERSISTENT RESERVE OUT command with the REGISTER or the REGISTER AND IGNORE EXISTING KEY service action and is still registered by any initiator, all RESERVE commands and all RELEASE commands regardless of initiator shall conflict and shall **report a reservation conflict condition**. If a logical unit has been reserved by any RESERVE command and is still reserved by any initiator, all PERSISTENT RESERVE IN and all PERSISTENT RESERVE OUT commands shall conflict regardless of initiator or service action and shall **report a reservation conflict condition**.

Sub-clause 5.5.1, Reservations overview, fifth paragraph from bottom of page 24, pdf page 43:

**conflict:** Commands issued by initiators not holding the reservation or by initiators not registered when a registrants only persistent reservation is present shall not be performed and the device server shall **report a reservation conflict condition**.

Sub-clause 5.5.3.4, Registering, third paragraph from bottom of page 30, pdf page 49:

Any PERSISTENT RESERVE OUT command service action received from an unregistered initiator, other than the REGISTER or the REGISTER AND IGNORE EXISTING KEY service action, shall **report a reservation conflict condition**.

Sub-clause 5.5.3.5, Creating a persistent reservation when there is no persistent reservation, third through sixth paragraphs from top of page 31, pdf page 50:

If the target receives a PERSISTENT RESERVE OUT command that attempts to create a persistent reservation when a persistent reservation already exists for the logical unit from an initiator other than the initiator that created the reservation, then the command shall **report a reservation conflict condition**.

If the initiator that created the persistent reservation attempts to modify the TYPE or SCOPE of an existing reservation, then the command shall **report a reservation conflict condition**.

If the device server receives a PERSISTENT RESERVE OUT command with a service action of RESERVE where the TYPE and SCOPE are the same as the existing TYPE and SCOPE from the initiator that created the persistent reservation, it shall not make any change to the existing reservation and shall return a GOOD status.

If the target receives a RESERVE(10) or RESERVE(6) command when a persistent reservation exists for the logical unit then the command shall **report a reservation conflict condition**.

Sub-clause 5.5.3.6.3.4, Removing registrations, third paragraph from top of page 36, pdf page 55:

If a PERSISTENT RESERVE OUT specifying a PREEMPT service action sets the SERVICE ACTION RESERVATION KEY field to a value that does not match any registered reservation key the device server shall **report a reservation conflict condition**.

Sub-clause 5.6, Multiple port and multiple initiator behavior, second paragraph, page 37, pdf page 56:

If a SCSI device has more than one service delivery port, the arbitration and connection management among the service delivery ports is vendor specific. If one service delivery port is being used by an initiator, accesses attempted through other service delivery port(s) may:

- a) **report a busy condition**; or
- b) be accepted as if the other service delivery port(s) were not in use.

Clause 6, Model for processor devices, fourth paragraph, page 39, pdf page 58:

If a processor device server has no resource available to manage a data packet from the application client, has no data packet to provide to the application client, or has no resources assigned to perform the operation, the device server may choose one of the following responses:

- a) Terminate the command with CHECK CONDITION status and the sense key NOT READY with the appropriate additional sense code for the condition. This is the appropriate response to a TEST UNIT READY command;
- b) Delay data transmission until the necessary resource or data packet becomes available;
- c) **Report a busy condition**; or
- d) Treat the logical unit as an incorrect logical unit (see SAM-2).

Clause 6, Model for processor devices, third paragraph from top of page 40, pdf page 59:

**conflict:** Commands issued by initiators not holding the reservation or by initiators not registered when a registrants only persistent reservation is present shall not be performed and the device server shall **report a reservation conflict condition**.

Sub-clause 7.10.4.3, Persistent Reservations Type, page 108, pdf page 127, table 72:

Code	Name	Description
0h		Obsolete
1h	Write Exclusive	Reads Shared: Any application client on any initiator may initiate tasks that request transfers from the storage medium or cache of the logical unit to the initiator.  Writes Exclusive: Any task from any initiator other than the initiator holding the persistent reservation that requests a transfer from the initiator to the storage medium or cache of the logical unit shall <b>report a reservation conflict condition</b> .
2h		Obsolete
3h	Exclusive Access	Reads Exclusive: Any task from any initiator other than the initiator holding the persistent reservation that requests a

		transfer from the storage medium or cache of the logical unit to the initiator shall <b>report a reservation conflict condition</b> . Writes Exclusive: Any task from any initiator other than the initiator holding the persistent reservation that requests a transfer from the initiator to the storage medium or cache of the logical unit shall <b>report a reservation conflict condition</b> .
4h		Obsolete
5h	Write Exclusive – Registrants Only	Reads Shared: Any application client on any initiator may initiate tasks that request transfers from the storage medium or cache of the logical unit to the initiator. Writes Exclusive: A task that requests a transfer to the storage medium or cache of the logical unit from an initiator that is not currently registered with the device server shall <b>report a reservation conflict condition</b> .
6h	Exclusive Access – Registrants Only	Reads Exclusive: A task that requests a transfer from the storage medium or cache of the logical unit to an initiator that is not currently registered with the device server shall <b>report a reservation conflict condition</b> . Writes Exclusive: A task that requests a transfer to the storage medium or cache of the logical unit from an initiator that is not currently registered with the device server shall <b>report a reservation conflict condition</b> .
7h – Fh		Reserved

Sub-clause 7.11.3, PERSISTENT RESERVE OUT parameter list, third paragraph from the bottom of page 111, pdf page 130:

Except as noted above, when a PERSISTENT RESERVE OUT command specifies a RESERVATION KEY field other than the reservation key registered for the initiator the device server shall **report a reservation conflict condition**. Except as noted above, the reservation key of the initiator shall be verified to be correct regardless of the SERVICE ACTION and SCOPE field values.

Sub-clause 7.21.2, Logical unit reservation, second paragraph, page 156, pdf page 175:

If the logical unit is reserved for another initiator, the device server shall **report a reservation conflict condition**.

Sub-clause 7.21.4, Superseding reservations, page 158, pdf page 177:

Superseding reservations is mandatory if the RELEASE(10) command is implemented. An application client that holds a current logical unit reservation may modify that reservation by issuing another RESERVE command to the same logical unit. The superseding RESERVE command shall release the previous reservation state when the new reservation request is granted. The current reservation shall not be modified if the superseding reservation request cannot be granted. If the superseding reservation cannot be granted because of conflicts with a previous reservation, other than the reservation being superseded, then the device server shall **report a reservation conflict condition**.

Sub-clause 7.25, TEST UNIT READY command, second paragraph, page 163, pdf page 182:

Table 117 defines the suggested GOOD and CHECK CONDITION status responses to the TEST UNIT READY command. Other conditions, including deferred errors, may result in other responses (e.g., **busy or reservation conflict conditions**).

Sub-clause 8.3.6, Control mode page, table 154, top of page 196, pdf page 215:

Define a reserved bit as RCINTLCK. I suggest bit 4 or 5 of byte 4.

Sub-clause 8.3.6, Control mode page, top of page 198, pdf page 217, add the following paragraphs:

A rejected command interlock (RCINTLCK) bit of zero specifies that the logical unit shall report busy conditions, reservation conflict conditions and task set full conditions by returning BUSY status, RESERVATION CONFLICT status or TASK SET FULL status respectively.

A rejected command interlock (RCINTLCK) bit of one specifies that the logical unit shall report busy conditions, reservation conflict conditions and task set full conditions by returning CHECK CONDITION status. The sense key shall be set to ABORTED COMMAND and the additional sense code shall be set to BUSY CONDITION REPORTED, RESERVATION CONFLICT CONDITION REPORTED or TASK SET FULL CONDITION REPORTED respectively.

Annex sub-clause B.1, SBC commands, last five paragraphs on page 229, pdf page 248:

**conflict:** Commands issued by initiators not holding the reservation or by initiators not registered when a registrants only persistent reservation is present shall not be performed and the device server shall **report a reservation conflict condition**.

Commands from initiators holding a reservation should complete normally. The behavior of commands from registered initiators when a registrants only persistent reservation is present is specified in table B.1, table B.2 and table B.3.

A command that does not explicitly write the medium shall be checked for reservation conflicts before the command enters the current task state for the first time. Once the command has entered the current task state, it shall not **report a reservation conflict condition** due to a subsequent reservation.

A command that explicitly writes the medium shall be checked for reservation conflicts before the device server modifies the medium or cache as a result of the command. Once the command has modified the medium, it shall not **report a reservation conflict condition** due to a subsequent reservation.

For each command, this standard, SPC-2, or a related command standard defines the conditions that **report a reservation conflict condition**. Depending on the particular command standard the conditions are defined in that standard's device model clause or in the subclauses that define the specific commands. An annex in SPC-2 contains the **reservation conflict** information for some of the command sets.

Annex sub-clause B.2, SMC commands, last five paragraphs on page 233, pdf page 252:

**conflict:** Commands issued by initiators not holding the reservation or by initiators not registered when a registrants only persistent reservation is present shall not be performed and the device server shall **report a reservation conflict condition**.

Commands from initiators holding a reservation should complete normally. The behavior of commands from registered initiators when a registrants only persistent reservation is present is specified in table B.4.

A command that does not explicitly write the medium shall be checked for reservation conflicts before the command enters the current task state for the first time. Once the command has entered the current task state, it shall not **report a reservation conflict condition** due to a subsequent reservation.

A command that explicitly writes the medium shall be checked for reservation conflicts before the device server modifies the medium or cache as a result of the command. Once the command has modified the medium, it shall not **report a reservation conflict condition** due to a subsequent reservation.

For each command, this standard, SPC-2, or a related command standard defines the conditions that result in RESERVATION CONFLICT. Depending on the particular command standard the conditions are defined in that standard's device model clause or in the subclauses that define the specific commands. An annex in SPC-2 contains the **reservation conflict** information for some of the command sets.