To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 7 August 2003
Subject: T10/03-271r0 SAM-3 SPC-3 Obsolete untagged tasks

Revision History

Revision 0 (7 August 2003) first revision

Related Documents

sam3r08 – SCSI Architecture Model – 3 revision 8
spc3r14 – SCSI Primary Commands – 3 revision 14

Overview

In SAM-1, transport protocols were required to specify a way to send untagged tasks.

In SAM-2, this was made optional. SRP, SPI-4 information unit mode, and SAS do not specify ways to send untagged tasks. FCP and iSCSI both overload their task attribute fields with an "untagged" value to implement this optional feature.

In SAM-3, untagged tasks should be made obsolete.

Suggested Changes to SAM-3

1 Scope

1.1 Introduction

…

The following architecture model concepts from previous versions of this standard are made obsolete by this standard:

a) Contingent Allegiance; ~~and~~

b) The TARGET RESET task management function; and

c) untagged tasks.~~.~~

3.1.42 I_T_L_Q nexus: A nexus between a SCSI initiator port, a SCSI target port, a logical unit, and a ~~tagged~~ task (see 4.12).
[Editor's note: changing "tagged task" to just "task" everywhere]

3.1.xxx tag: A field containing up to 64 bits that is a component of an I_T_L_Q nexus. See 4.11.2.
[Editor's note: changing "task tag" to just "tag" everywhere. There are only a few uses of "task tag".]

4.4 The SCSI structural model

Figure 8 – Overall SCSI domain model
[Delete the Untagged Task box. Change Tagged Task box to Task.]

4.8 Logical units

A logical unit (see figure 14) contains:

a) A logical unit number;
    A) If access controls (see SPC-3) are not in effect, one logical unit number per logical unit; or
    B) If access controls are in effect, one logical unit number per SCSI initiator port that has access rights plus one default logical unit number per logical unit;
b) One ore more logical unit names;
c) A device server;
d) A task manager; and

e) One or more task sets each of which may contain ~~zero or more untagged tasks or a combination of~~ zero or more ~~tagged~~ tasks ~~and zero or more untagged tasks~~.

Figure 14 — Logical unit model
[Delete the Untagged Task box. Change Tagged Task box to Task.]
…

30 working draft SCSI Architecture Model - 3 (SAM-3)
A task set is composed of ~~zero or more untagged tasks or a combination of~~ zero or more ~~tagged~~ tasks (see 4.11) ~~and zero or more untagged tasks~~. ~~See 4.11 for additional restrictions on the untagged tasks and tagged tasks in a task set.~~

~~Task (see 4.11) refers to either a tagged task or an untagged task.~~ The interactions among the tasks in a task set are determined by the requirements for task set management specified in clause 8 and the ACA requirements specified in 5.9.1. The number of task sets per logical unit and the boundaries between task sets are governed by the TST field in the Control mode page (see SPC-2).

4.11 Tasks

4.11.1 The task object

The task object represents ~~either~~ a ~~tagged~~ task ~~or an untagged task~~. The composition of a task includes a definition of the work to be performed by the logical unit in the form of a command or a group of linked commands.

A ~~tagged~~ task is represented by an I_T_L_Q nexus (see 4.12) and is composed of a definition of the work to be performed by the logical unit~~,~~ and a task attribute (see 8.5). ~~An untagged task is represented by an I_T_L nexus (see 4.12) and is composed of a definition of the work to be performed by the logical unit, and implicitly a SIMPLE task attribute (see 8.5).~~

The I_T_L_Q nexus representing a ~~tagged~~ task includes a tag (see 4.11.2) allowing many uniquely identified ~~tagged~~ tasks to be present concurrently in a single task set. A ~~tagged~~ task also includes one of the task attributes described in 8.5 that allows the application client to specify processing relationships between various ~~tagged~~ tasks.

~~An untagged task does not include a tag in its I_T_L nexus, thus restricting the number of concurrent untagged tasks in a single task set to one per SCSI initiator port. An untagged task is assumed to have a SIMPLE task attribute.~~

~~Every SCSI transport protocol shall support tagged tasks and may support untagged tasks. If the SCSI transport protocol upon which a SCSI device operates supports untagged tasks, the SCSI device is not required to support tagged tasks.~~

An I_T_L_x nexus that is in use (i.e., during the interval bounded by the events specified in 5.5) shall be unique as seen by the SCSI initiator port originating the command and the logical unit to which the command was addressed~~,~~; otherwise, an overlapped command condition exists (see 5.9.3). An I_T_L_x nexus is unique if one or more of its components is unique within the specified time interval. ~~An untagged task shall be unique with respect to all tagged tasks in the task set.~~

~~A SCSI initiator device shall not cause the creation of more than one untagged task from a specific SCSI initiator port having identical values for the target port identifier and logical unit number.~~ A SCSI initiator device shall not create more than one task from a specific SCSI initiator port having identical values for the target port identifier, logical unit number, and tag.

4.11.2 ~~Task tags~~Tag

A tag is a field containing up to 64 bits that is a component of an I_T_L_Q nexus. A SCSI initiator device assigns tag values in each I_T_L_Q nexus in a way that ensures that the nexus uniqueness requirements stated in 4.11.1 are met. Transport protocols may define restrictions on tag assignment (e.g., tags may be required to be unique per I_T nexus or per I_T_L nexus).

4.12 The nexus object

The nexus object represents a relationship between a SCSI initiator port, a SCSI target port, optionally a logical unit~~.~~ , and optionally a task.

The nexus object may refer to any one or all of the following relationships:

a) One SCSI initiator port to one SCSI target port (an I_T nexus);
b) One SCSI initiator port to one SCSI target port to one logical unit (an I_T_L nexus);
c) One SCSI initiator port to one SCSI target port to one logical unit to one ~~tagged~~ task (an I_T_L_Q nexus); or
d) Either an I_T_L nexus or an I_T_L_Q nexus (denoted as an I_T_L_x nexus).

Table 19 — Mapping nexus to SAM-2 identifiers
Tag 4.11.2

4.14 Model for dependent logical units

…

Figure 23 — Dependent logical unit model
[Delete the Untagged Task box. Change Tagged Task box to Task.]

5.1 The Execute Command procedure call

…

Task Attribute: A value specifying one of the task attributes defined in 8.5. This argument shall not be specified for ~~an untagged command or~~ the second and subsequent commands in a sequence of linked commands. ~~Untagged tasks shall implicitly have the SIMPLE attribute.~~ The attribute of a task that processes linked commands shall be set according to the Task Attribute argument specified for the first command in the sequence.

…

5.3.1 Status codes

…

TASK SET FULL. This status shall be implemented ~~if the logical unit supports the creation of tagged tasks (see 4.11). This status shall not be implemented if the logical unit does not support the creation of tagged tasks~~by all logical units.

When the logical unit has at least one task in the task set for a SCSI initiator port and a lack of task set resources prevents accepting a received ~~tagged~~ task from that SCSI initiator port into the task set, TASK SET FULL shall be returned. When the logical unit has no task in the task set for a SCSI initiator port and a lack of task set resources prevents accepting a received ~~tagged~~ task from that SCSI initiator port into the task set, BUSY should be returned.

~~When the logical unit has at least one task in the task set and a lack of task set resources prevents accepting a received untagged task into the task set, BUSY should be returned.~~

The logical unit should allow at least one command in the task set for each supported SCSI initiator port that has identified itself to the SCSI target port by a SCSI transport protocol specific ~~procedure call~~manner or by the successful transmission of a command.

If the UA_INTLCK_CTRL field in the Control mode page contains 11b (see SPC-3), termination of a command with TASK SET FULL status shall cause an unit attention condition to be established for the SCSI initiator port that sent the command with an additional sense code of PREVIOUS TASK SET FULL STATUS unless a PREVIOUS TASK SET FULL STATUS unit attention condition already exists.

7 Task management functions

7.1 Introduction

…

I_T_L_Q Nexus: A SCSI initiator port, SCSI target port, logical unit, and tag nexus (see 4.12).

7.2 ABORT TASK

This function shall be supported by all logical units~~ if it supports tagged tasks and may be supported by a logical unit if it does not support tagged tasks~~.

7.5 CLEAR TASK SET

~~This function shall be supported by all logical units, except in the following cases, when support for this function is optional:~~
~~a) The logical unit does not support tagged tasks (see 4.11); or~~
~~b) The logical unit supports the basic task management model (see 8.2).~~
This function may be supported by logical units supporting the basic task management model (see 8.2) and shall be supported by all other logical units.

7.8 Task management SCSI transport protocol services

Since the nexus used by all task management functions except ABORT TASK and QUERY TASK does not contain a ~~task~~ tag to uniquely identify the transaction, there may be no way for an application client to associate a confirmation with a request. A SCSI transport protocol that does not provide such an association should not allow a SCSI initiator port to have more than one pending task management request per I_T_L nexus.

8 Task Set Management

8.2 ~~Controlling task set management~~Task management models

8.2.1 Task management model overview

The Control mode page (see SPC-2) contains fields that specify particular task set management behaviors. The standard INQUIRY data ~~CmdQue~~CMDQUE bit and BQUE bit (see SPC-~~2~~3) indicate~~s~~ support for ~~tagged tasks (command queuing). One specific combination of task set management behaviors is identified as~~ the basic or full task management model~~s~~.

~~Support for the basic task management model is indicated by values returned in the CMDQUE and BQUE bits in the standard INQUIRY data (see SPC-2).~~

8.2.2 Basic task management model

The basic task management model requires the following task set management behaviors:

a) The only task attribute supported shall be SIMPLE;
b) The device server may reorder the actual processing sequence of tasks in any manner. Any data integrity exposures related to task sequence order shall be explicitly handled by the application client using the appropriate commands (i.e., they shall be handled as if the QUEUE ALGORITHM MODIFIER field in the Control mode page (see SPC-3) is set to 1h);
c) All the tasks in the task set shall be aborted when a CHECK CONDITION status is returned~~ or when an ACA condition is established~~ for any task (i.e., they shall be handled as if the QERR field in the Control mode page (see SPC-3) is set to 01b);
~~d) It shall not be possible to disable tagged queuing;~~ and
e) Support for the CLEAR TASK SET task management function is optional.
[Question: should the basic model also assume that TST=0? Or is a “basic” logical unit allowed to have a separate task set per initiator? Assuming not…]

8.2.3 Full task management model

The full task management model requires the following task set management behaviors:

a) Task attributes other than SIMPLE may be supported;

b) The QUEUE ALGORITHM MODIFIER field in the Control mode page (see SPC-3) shall control the processing sequence of tasks having the SIMPLE task attribute;

c) The QERR field in the Control mode page (see SPC-3) shall control aborting of tasks when a CHECK CONDITION status is returned for any task; and

d) support for the CLEAR TASK SET task management function is mandatory.

Suggested changes to SPC-3

3.1.111 task set: A group of tasks within a logical unit, whose interaction is dependent on the task management ~~(queuing)~~ and ACA rules. See SAM-2 and the Control mode page (see 7.4.6).

3.1.19 Control mode page: A mode page that provides controls over several SCSI features (e.g., ~~tagged queuing and~~ error logging) that are applicable to all device types. See 7.4.6.

5.5.1 Reservations overview

…

Multiple persistent reserve service actions may be present in the task set at the same time. The order of execution of such service actions is defined by the ~~tagged queuing restrictions, if any~~task set management requirements (see SAM-3), but each is executed as a single indivisible command without any interleaving of actions that may be required by other reservation commands.

5.5.2.7.4.2 Failed persistent reservation preempt

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

6.4.2 Standard INQUIRY data

…

~~When the CMDQUE bit is set to zero, the BQUE bit shall have the following meaning. A BQUE bit set to zero indicates that the device does not support tagged tasks (command queuing) for this logical unit. A BQUE bit set to one indicates that the device supports, for this logical unit, the basic task management model defined by SAM-2.~~

~~…~~

~~A command queuing (CMDQUE) bit set to one indicates that the device supports tagged tasks (command queuing) for this logical unit (see SAM-2). A CMDQUE bit set to zero indicates the device server may support tagged tasks for this logical unit (see the BQUE bit, above). Table 79 summarizes the relationship of the BQUE and CMDQUE bits.~~

The command queuing (CMDQUE) and basic queueing (BQUE) bits indicate whether the logical unit supports the full task management model or the basic task management model as described in table 79.

Table 79 — Relationship of BQUE and CMDQUE bits

BQUE	CMDQUE	Description
0	0	~~No command queuing of any kind supported.~~Obsolete.
0	1	~~Command queuing with all types of task tags supported.~~Full task management model supported (see SAM-3).
1	0	Basic task management model supported (see SAM-~~2~~3).
1	1	Illegal combination of BQUE and CMDQUE bits.

7.4.6 Control mode page

The Control mode page (see table 223) provides controls over several SCSI features that are applicable to all device types such as ~~tagged queuing and~~ error logging.

…

A disable queuing (DQUE) bit set to zero specifies that ~~tagged queuing shall be enabled~~ multiple tasks shall be allowed in the task set~~if the device server supports tagged queuing~~. A DQUE bit set to one specifies that ~~tagged queuing shall be disabled~~only one task shall be allowed in the task set~~. Any~~ and any ~~queued~~ commands ~~received by the device server~~ currently in the task set other than the MODE SELECT command shall be aborted. ~~The method used to abort queued commands is protocol specific.~~