To: T10 Technical Committee
From: Rob Elliott, HP (elliott@hp.com)
Date: 22 January 2004
Subject: 03-351r2 SAM-3 SPC-3 Reporting task attribute support

Revision history
Revision 0 (14 October 2003) First revision
Revision 1 (11 December 2003) Incorporated comments from November 2003 CAP WG - removed the enable/disable feature requested for multipathing software.
Revision 2 (22 January 2004) Incorporated comments from January 2004 CAP WG.

Related documents
sam3r10 - SCSI Architecture Model - 3 revision 10
spc3r16 - SCSI Primary Commands - 3 revision 16
03-388 - SBC-2 Non-volatile caches (Rob Elliott, HP)

Overview

There are currently four task attributes defined in SAM-3:

a) SIMPLE;
b) ORDERED;
c) HEAD OF QUEUE; and
d) ACA.

Problems:

1. There is no way for an application client to tell which task attributes are supported. Some logical units may prefer not to support ORDERED, for example. This proposal lets software read a VPD page to determine which task attributes are supported and which are not supported. It also clarifies what happens if an unsupported task attribute is requested.

2. In SBP-3, unlike most other protocols, there is no task attribute sent with each command. A logical unit is configured to treat all commands as SIMPLE or all commands as ORDERED. This proposal lets SCSI software determine which policy is in place.

As suggested in the January 2004 CAP WG, this proposal takes over the (also new) Protection Information VPD page, renaming it Extended INQUIRY Data and extending it to 64 bytes. If the page is supported, the 3 new bits must be implemented (otherwise they cannot reliably indicate non-support for a task attribute).

Suggested changes to SAM-3

5 SCSI command model

5.1 The Execute Command procedure call

An application client requests the processing of a SCSI command by invoking the SCSI transport protocol services described in 5.4, the collective operation of which is conceptually modeled in the following procedure call:

> Service Response =Execute Command (IN (I_T_L_Q Nexus, CDB, Task Attribute, [Data-In Buffer Size], [Data-Out Buffer], [Data-Out Buffer Size], [Command Reference Number]), OUT ([Data-In Buffer], [Sense Data], [Sense Data Length], Status))

Input Arguments:

> **I_T_L_Q Nexus**: The I_T_L_Q nexus identifying the task (see 4.12).
> **CDB**: Command descriptor block (see 5.2).
> **Task Attribute**: A value specifying one of the task attributes defined in 8.6. Transport protocols may or may not provide the ability to specify a different task attribute for each task (see 8.6.0). For a task that processes linked commands, the Task Attribute shall be that specified for the first command in the sequence of linked commands. The Task Attribute specified for the second and subsequent commands shall be ignored.

5.9.5 Task attribute exception conditions

~~When an attempt to create a task with an invalid task attribute is detected, the command shall be terminated~~ If a device server receives a command with a task attribute that is not supported or is not valid (e.g., an ACA task attribute when an ACA condition does not exist), it shall terminate the command with CHECK CONDITION status, sense key of ILLEGAL REQUEST and additional sense code of INVALID MESSAGE ERROR. Task attribute support should be reported with the Extended INQUIRY Data VPD page (see SPC-3).

NOTE 1 (really 12) - The use of the INVALID MESSAGE ERROR additional sense code is based on its similar usage in previous versions of this standard. The use of the INVALID MESSAGE ERROR additional sense code is not to be interpreted as a description of how the task attributes are represented by any given SCSI transport protocol.

8.2 Implicit head of queue

A command standard (see 3.1.15) may define tasks that may be processed by the task manager as if the task's task attribute is HEAD OF QUEUE without regard to the actual task attribute received with the task.

8.3 Controlling task set management

The Control mode page (see SPC-2) contains fields that specify particular task set management behaviors. The standard INQUIRY data CMDQUE bit (see SPC-2) indicates support for tagged tasks (command queuing). One specific combination of task set management behaviors is identified as the basic task management model. 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). The basic task management model requires the following task set management behaviors:

8.3.2 Full task management model

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

a) The SIMPLE task attribute (see 8.6.1) shall be supported;
b) Task attributes other than SIMPLE may be supported;
c) 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;
d) 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
e) The CLEAR TASK SET task management function (see 7.5) shall be supported.

8.3.3 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~~ The task manager only supports the SIMPLE task attribute or the task manager only supports the ORDERED task attribute (see 8.6.1);
b) The ~~device server~~ task manager 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 tasks in the task set shall be aborted when a CHECK CONDITION status is returned 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) If the Control mode page is supported, the QUEUE ALGORITHM MODIFIER field shall be set to 1h and the QERR field shall be set to 01b and they shall not be changeable; and
e) The CLEAR TASK SET task management function (see 7.5) may be supported.

8.6 Task attributes

8.6.0 Task attribute overview

The application client shall assign each task one of the task attributes listed in table 1.

Table 1 — Task attributes [new table]

Task attribute	Reference
SIMPLE	8.6.1
ORDERED	8.6.2
HEAD OF QUEUE	8.6.3
ACA	8.6.4

Transport protocols shall either:

a) provide the capability to specify a unique task attribute for each task; or
b) require use of the basic task management model (see 8.3.3).

Transport protocols should provide the capability to specify a unique task attribute for each task.

8.6.1 Simple task

If accepted, a~~A~~ task having the SIMPLE attribute shall be ~~accepted~~entered into the task set in the dormant task state. The task shall not enter the enabled task state until all older head of queue tasks and older ordered tasks in the task set have ended (see 8.4).

The QUEUE ALGORITHM MODIFIER field in the Control mode page (see SPC-2) provides additional constraints on task completion order for tasks with the SIMPLE attribute.

8.6.2 Ordered task

If accepted, a~~A~~ task having the ORDERED attribute shall be ~~accepted~~entered into the task set in the dormant task state. The task shall not enter the enabled task state until all older tasks in the task set have ended (see 8.4).

8.6.3 Head of queue task

If accepted, a~~A~~ task having the HEAD OF QUEUE attribute shall be ~~accepted~~entered into the task set in the enabled task state.

8.6.4 ACA task

If accepted, a~~A~~ task having the ACA attribute shall be ~~accepted~~entered into the task set in the enabled task state. There shall be no more than one ACA task per task set (see 5.9.2.1).

Suggested changes to SPC-3

6.4.2 Standard INQUIRY data

...

The Normal ACA Supported (NORMACA) bit set to one indicates that the device server supports setting the NACA bit to one in the CONTROL byte of the CDB and supports the ACA task attribute (see SAM-2). A NORMACA bit set to zero indicates that the device server does not support ~~setting the~~a NACA bit set to one and does not support the ACA task attribute.

7.6 Vital product data parameters

7.6.1 Vital product data parameters overview and page codes

This subclause describes the vital product data (VPD) page structure and the VPD pages (see table 268) that are applicable to all SCSI devices. These VPD pages are optionally returned by the INQUIRY command (see 6.4) and contain vendor specific product information about a target or logical unit. The vital product data may include vendor identification, product identification, unit serial numbers, device operating definitions,

manufacturing data, field replaceable unit information, and other vendor specific information. This standard defines the structure of the vital product data, but not the contents.

Table 2 — Vital product data page codes

Page code	VPD page name	Reference	Support requirements
...	...		
86h	~~Protection Information~~Extended INQUIRY Data	7.6.6	Optional
...	...		

Editor's Note 1: Also change in annex C.6. It should be available to all device types.

7.6.6 ~~Protection Information~~ Extended INQUIRY Data VPD page

The ~~Protection Information~~ Extended INQUIRY Data VPD page (see table 298) provides the application client with the means to obtain ~~protection information (see SBC-2)~~ miscellaneous read-only parameters supported by the logical unit.

Table 3 — ~~Protection Information~~Extended INQUIRY Data VPD page

Byte\Bit	7	6	5	4	3	2	1	0
0	PERIPHERAL QUALIFIER			PERIPHERAL DEVICE TYPE				
1	PAGE CODE (86h)							
2	Reserved							
3	PAGE LENGTH (~~4~~60)							
4	Reserved					GARD_CHK	ATPG_CHK	RFTG_CHK
5	Reserved					HEADSUP	ORDSUP	SIMPSUP
~~5~~6	Reserved							
~~7~~63								

The PERIPHERAL QUALIFIER field and the PERIPHERAL DEVICE TYPE field are as defined in 6.4.2.

The PAGE LENGTH field specifies the length of the following VPD page data and shall be set to four. If the allocation length is less than the length of the data to be returned, the page length shall not be adjusted to reflect the truncation.

A guard check (GARD_CHK) bit set to zero indicates the device server does not check the LOGICAL BLOCK GUARD field in the protect~~ed~~ion information (see SBC-2) before transmitting it to an application client. A GARD_CHK bit set to one indicates the device server checks the LOGICAL BLOCK GUARD field in the protect~~ed~~ion information before transmitting it to an application client. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field containing FFFFh, the checking of the LOGICAL BLOCK GUARD field in the protect~~ed~~ion information shall not be performed for the associated logical block.

An application tag check (APTG_CHK) bit set to zero indicates the device server does not check the LOGICAL BLOCK APPLICATION TAG field in the protect~~ed~~ion information (see SBC-2) before transmitting it to an application client. An APTG_CHK bit set to one indicates the device server checks the LOGICAL BLOCK APPLICATION TAG field in the protect~~ed~~ion information before transmitting it to an application client. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field containing FFFFh, the checking of the LOGICAL BLOCK APPLICATION TAG field in the protect~~ed~~ion information shall not be performed for the associated logical block.

A reference tag check (RFTG_CHK) bit set to zero indicates the device server does not check the LOGICAL BLOCK REFERENCE TAG field in the protect~~ed~~ion information (see SBC-2) before transmitting it to an application client. A RFTG_CHK bit set to one indicates the device server checks the LOGICAL BLOCK REFERENCE TAG field in the protect~~ed~~ion information before transmitting it to an application client. If the application client or device server detects a LOGICAL BLOCK APPLICATION TAG field containing FFFFh, the checking of the LOGICAL BLOCK REFERENCE TAG field in the protect~~ed~~ion information shall not be performed for the associated logical block.

A HEAD OF QUEUE supported (HEADSUP) bit set to one indicates the HEAD OF QUEUE task attribute is supported by the logical unit. A HEADSUP bit set to zero indicates the HEAD OF QUEUE task attribute is not supported.

An ORDERED supported (ORDSUP) bit set to one indicates the ORDERED task attribute is supported by the logical unit. An ORDSUP bit set to zero indicates the ORDERED task attribute is not supported.

A SIMPLE supported (SIMPSUP) bit set to one indicates the SIMPLE task attribute is supported by the logical unit. A SIMPSUP bit set to zero indicates the SIMPLE task attribute is not supported.

Application clients should not use a task attribute if it is reported as not supported in the HEADSUP bit, ORDSUP bit, or SIMPSUP bit. SAM-3 defines handling of unsupported task attributes. If this VPD page is implemented, the HEADSUP bit, ORDSUP bit, and SIMPSUP bit shall indicate if the corresponding task attributes are supported.

Editor's Note 2: 03-388 also plans to add bits to this page indicating non-volatile cache support.
