T10/07-157 revision 0

Date: May 03, 2007

To: T10 Committee (SCSI)

From: George Penokie (IBM)

Subject: SAM-4: Changes requested from 03/2007 editing session

1 Overview

During the SAM-4 editing session in March it was requested that the UML diagrams be enhanced to include all the parameters that are listed in the Execute Command procedure call and to add in associations between some of the classes in the SCSI initiator device and the logical unit class diagrams. This proposal makes those changes.

4.5.9 SCSI initiator device class

A SCSI initiator device class (see figure 9) is a SCSI device class that contains the:

a) ~~A SCSI initiator device~~ application client class ~~(see figure 19)~~(see 4.5.10) ~~is a SCSI device class~~ that contains the:
   A) application client task class ~~(see 4.5.10) that contains the:~~(see 4.5.12); and
   B) application ~~client task~~ buffer manager class ~~(see 4.5.11)~~(see 4.5.11).

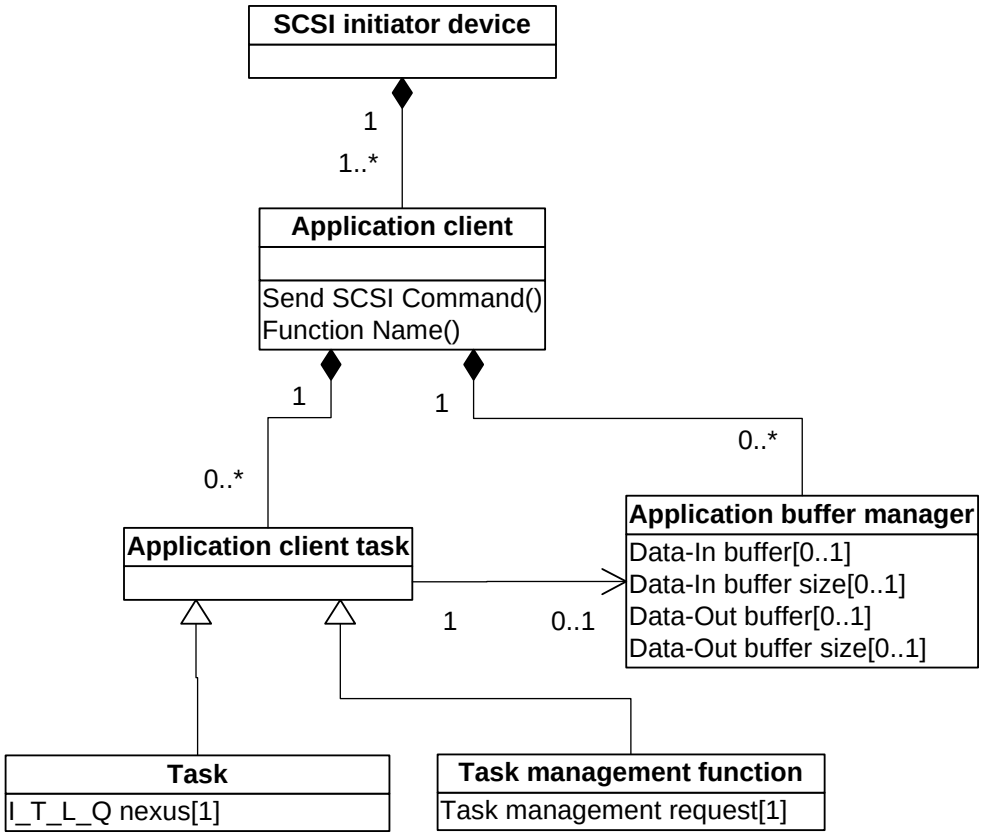

Figure 1 — SCSI initiator device class diagram

Each instance of a SCSI initiator device class shall contain the following objects:

a) one or more application clients that contain:
   A) ~~one~~ zero or more application ~~clients that contain:~~client tasks; and

B) zero or more application client ~~tasks~~buffer managers.

4.5.10 Application client class

~~An application client class contains zero or more application client tasks.~~

An application client class contains zero or more application client tasks and for each application client task zero or one application client buffer manager.

An application client class originates commands by issuing a **Send SCSI Command** requests ~~(see 5.4.2)~~(see 5.4.2).

An application client class originates task management requests by issuing a Function name service request ~~(see clause 7)~~(see clause 7).

An application client may request processing of a task management function through a request directed to the task manager within the logical unit. The interactions between the task manager and application client when a task management request is processed are shown in ~~7.13~~7.13.

4.5.11 Application ~~client task~~ buffer management class

4.5.11.1 Application buffer management class overview

An application buffer management class manages any data-in buffers or data-out buffers that occur as a result of an application client class originating a command that requirement buffer management.

4.5.11.2 Data-In buffer attribute

The data-in buffer attribute contains the Data-In Buffer argument from an Execute Command procedure call (see 5.1).

4.5.11.3 Data-In buffer size attribute

The data-in buffer size attribute contains the Data-In Buffer Size argument from an Execute Command procedure call (see 5.1).

4.5.11.4 Data-Out buffer attribute

The data-out buffer attribute contains the Data-Out Buffer argument from an Execute Command procedure call (see 5.1).

4.5.11.5 Data-Out buffer size attribute

The data-out buffer size attribute contains the Data-Out Buffer Size argument from an Execute Command procedure call (see 5.1).

4.5.12 Application client task class

An application client task class (see figure 9) shall be substituted with:

a) a task class (see 4.5.14); or
b) a task management function class (see 4.5.13).

An application client task class is the source for a single command or a single task management function.

If the task class substituted by the application client task class requires buffer management, then the application client task class shall use the application client buffer manager for the buffer management.

4.5.13 Task management function class

4.5.13.1 Task ~~class~~ management function overview

A task management function class is an application client class that represents a ~~command (see clause 5)~~SCSI task management function (see clause 7).

4.5.13.2 Task management request attribute

The task management request attribute contains a task management request (see clause 7).

4.5.14 Task class

4.5.14.1 Task class overview

A task class (see figure 10) contains the:

a) task request class (see 4.5.15);
b) task indication class (see 4.5.16);
c) task response class (see 4.5.17); and
d) task confirmation class (see 4.5.18).

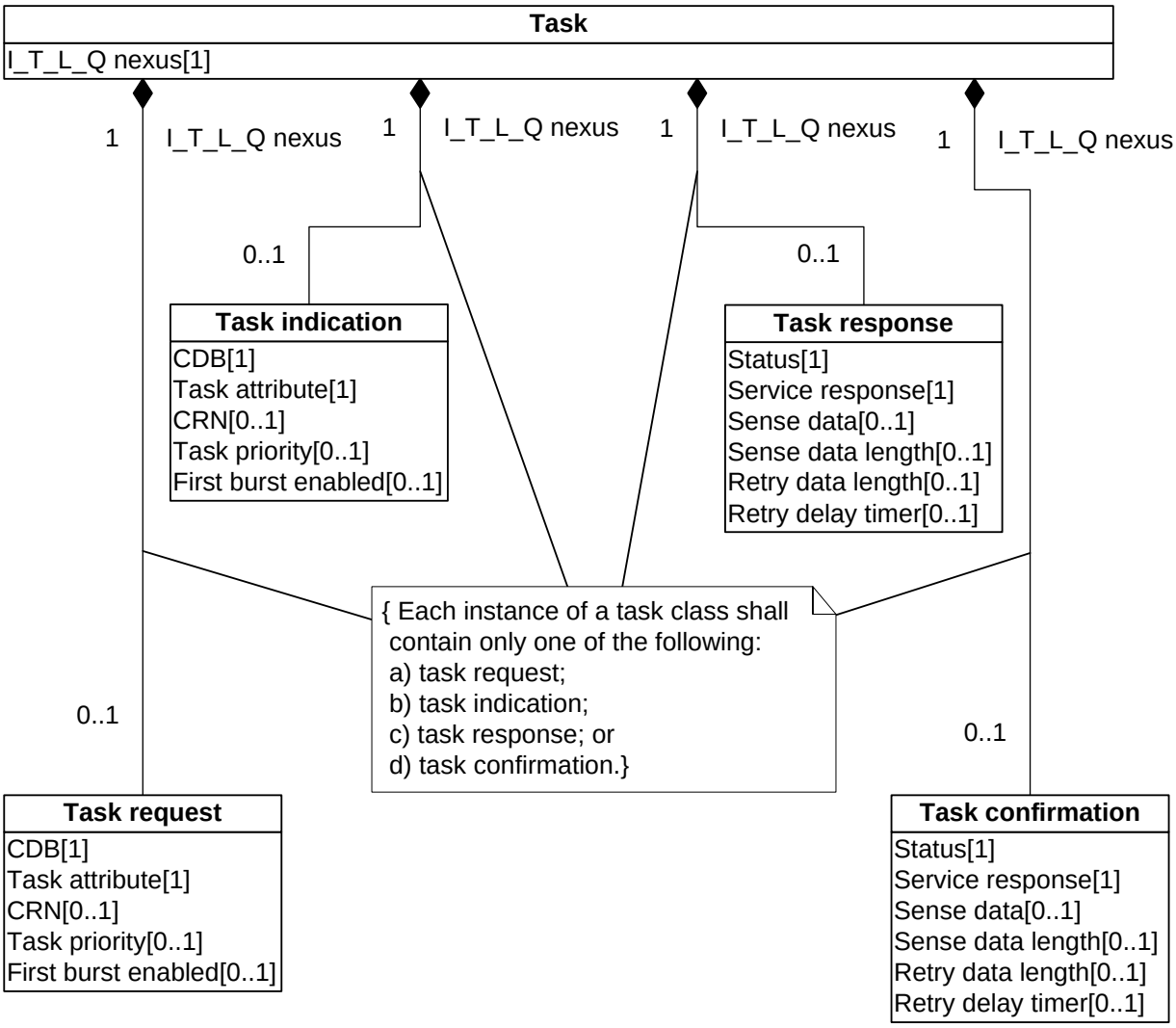

Figure 2 — Task class diagram

Each instance of a task class shall contain only one of the following objects:

a) one task request;
b) one task indication;
c) one task response; or
d) one task confirmation.

~~Each instance of a task class represents the work associated with a command.~~ A new command causes the creation of a ~~task~~task class. The task class persists until a ~~task~~ command complete ~~response~~ confirmation is ~~sent~~ received or until the task is ended by a task management function or exception condition. For an example of the processing for a command see ~~5.7~~5.7.

A task is represented by an I_T_L_Q nexus (see 4.7).

Each instance of a task class represents the work associated with a command.

4.5.14.2 I_T_L_Q nexus attribute

The I_T_L_Q nexus attribute contains the I_T_L_Q nexus of the task (see 4.7).

4.5.15 Task request class

4.5.15.1 Task request class overview

The task request class delivers a command to the service delivery subsystem.

4.5.15.2 CDB attribute

The CDB attribute contains a CDB (see 5.2 and SPC-3) that defines the work to be performed by a logical unit.

4.5.15.3 ~~CDB~~ Task attribute

A task attribute (see 8.6) identifies a command. The I_T_L_Q nexus representing a task includes a task attribute, allowing many uniquely identified tagged tasks to be present in a single task set. A task attribute is composed of up to 64 bits.

~~The CDB attribute contains a CDB (see 5.2 and SPC-3).~~

A SCSI initiator device assigns task attribute values for each I_T_L_Q nexus in a way that ensures that the nexus uniqueness requirements stated in this subclause are met. Transport protocols may define additional restrictions on task attribute assignment (e.g., restricting task attribute length, requiring task attribute to be unique per I_T nexus or per I_T_L nexus, or sharing task attribute values with other uses such as task management functions).

An I_T_L_Q 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 5.8.3). An I_T_L_Q nexus is unique if one or more of its components is unique within the specified time interval.

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 task attribute.

4.5.15.4 CRN attribute

The CRN attribute contains the CRN of the command (see 5.4.2.2).

4.5.15.5 Task priority attribute

The task priority attribute contains the priority of the command (see 8.7).

4.5.15.6 First burst enabled attribute

The first burst enabled attribute contains specifies if first burst for the command is enabled (see 5.4.2.2).

4.5.16 Task indication class

4.5.16.1 Task indication class overview

The task indication class receives a command from the service delivery subsystem.

4.5.16.2 Task indication attributes

For a description of the task indication attributes see 4.5.15.

4.5.17 Task ~~management function~~ response class

4.5.17.1 Task ~~management function~~ response class overview

~~A task management function class is an application client class that represents a SCSI task management function (see clause 7).~~

~~**4.5.17.2 Task management request attribute**~~

The task response class sends the response to a command to the service delivery subsystem.

4.5.17.3 Status attribute

The status attribute contains the status of the completed command (see 5.3).

4.5.17.4 Service response attribute

The service response attribute contains the service response for the completed command (see 5.4.2.5)

4.5.17.5 Sense data attribute

The ~~task management request~~ sense data attribute contains ~~a task management request (see clause 7)~~the sense data for the completed command (see 5.4.2.5).

4.5.17.6 Sense data length attribute

The sense data length attribute contains the length of the sense data for the completed command (see 5.4.2.5)

4.5.17.7 Retry delay timer attribute

The retry delay timer attribute contains the retry delay time for the completed command (see 5.4.2.5)

4.5.18 Task confirmation class

4.5.18.1 Task confirmation class overview

The task confirmation class receives the response to a command from the service delivery subsystem.

4.5.18.2 Task confirmation attributes

For a description of the task confirmation attributes see 4.5.17.

4.5.19 Logical unit class

4.5.19.1 Logical unit class overview

A logical unit class (see figure 13) contains the:

a) device server class (see 4.5.25);
b) task manager class (see 4.5.26); and
c) task set class (see 4.5.27).

A logical unit class (see figure 13) may be substituted with the:

a) well known logical unit class (see 4.5.24.1); or
b) hierarchical logical unit class.

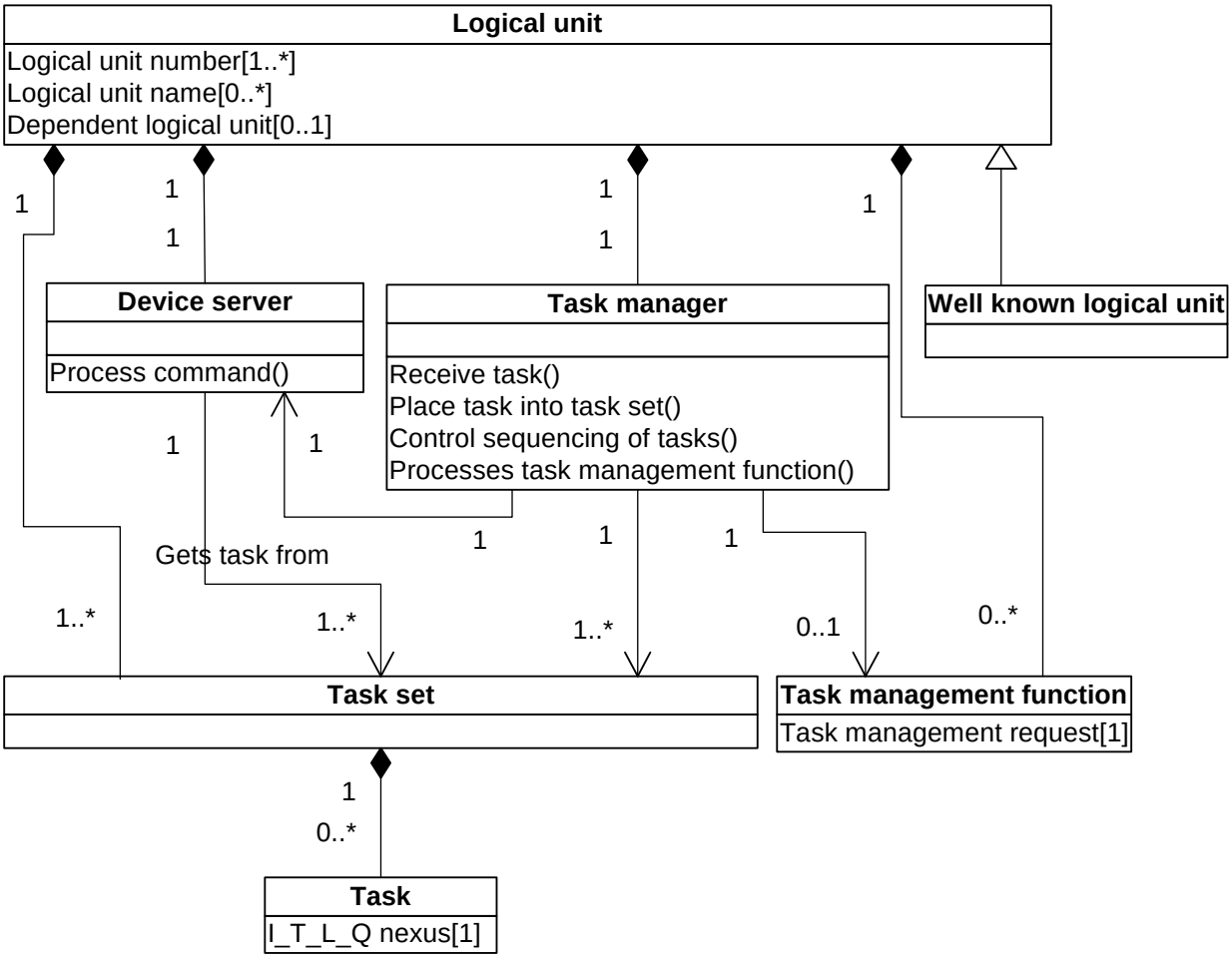

Figure 3 — Logical unit class diagram

Each instance of a logical unit class shall contain the following objects:

a) one device server;
b) one task manager; and
c) one or more task sets.

A logical unit is the class to which commands are sent. One of the logical units within the SCSI target device shall be accessed using the logical unit number zero or the REPORT LUNS well-known logical unit number.

If the logical unit inventory changes for any reason (e.g., completion of initialization, removal of a logical unit, or creation of a logical unit), then the device server shall establish a unit attention condition ~~(see 5.8.7)~~(see 5.8.7) for the initiator port associated with every I_T nexus, with the additional sense code set to REPORTED LUNS DATA HAS CHANGED.

4.5.19.2 Logical unit number attribute

A logical unit number attribute identifies the logical unit within a SCSI target device when accessed by a SCSI target port. If any logical unit within the scope of a SCSI target device includes one or more dependent logical units (see 4.5.24.4) in its composition, then all logical unit numbers within the scope of the SCSI target device shall have the format described in 4.6.6. If there are no dependent logical units within the scope of the SCSI target device, the logical unit numbers should have the format described in 4.6.5.

The 64-bit quantity called a LUN is the logical unit number attribute defined by this standard. The fields containing the acronym LUN that compose the logical unit number attribute are historical nomenclature

anomalies, not logical unit number attributes. Logical unit number attributes having different values represent different logical units, regardless of any implications to the contrary in 4.6 (e.g., LUN 00000000 00000000h is a different logical unit from LUN 40000000 00000000h and LUN 00FF0000 00000000h is a different logical unit from LUN 40FF0000 00000000h).

Logical unit number(s) are required as follows:

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.

See 4.6 for a definition of the construction of logical unit numbers to be used by SCSI target devices. Application clients should use only those logical unit numbers returned by a REPORT LUNS command. The task router shall respond to logical unit numbers other than those returned by a REPORT LUNS command (i.e., incorrect logical unit numbers) as specified in ~~5.8.4~~5.8.4 and ~~7.12~~7.12.

4.5.19.3 Logical unit name attribute

A logical unit name attribute identifies a name ~~(see 3.1.71)~~(see 3.1.71) for a logical unit that is not a well known logical unit. A logical unit name shall be world wide unique. A logical unit name shall never change and may be used to persistently identify a logical unit.

Logical unit name(s) are required as follows:

a) one or more logical unit names if the logical unit is not a well-known logical unit; or
b) zero logical unit names in the logical unit is a well-known logical unit.

4.5.19.4 Dependent logical unit attribute

A dependent logical unit attribute identifies a logical unit that is addressed via a hierarchical logical unit that resides at a lower numbered level in the hierarchy (i.e., no logical unit within level 1 contains a dependent logical unit attribute while all logical units within level 2, level 3, and level 4 do contain a dependent logical unit attribute).

Any instance of a logical unit class that contains dependent logical unit attribute shall utilize the hierarchical logical unit number structure defined in 4.6.6. If any logical unit within a SCSI target device includes dependent logical unit attribute:

a) all logical units within the SCSI target device shall format all logical unit numbers as described in 4.6.6; and
b) logical unit number zero or the REPORT LUNS well-known logical unit (see SPC-3) shall set the HISUP bit to one in the standard INQUIRY data.

4.5.20 Device server class

The device server class processes commands.

4.5.21 Task manager class

The task manager class:

a) receive tasks from a task router;
b) place tasks into a task set;
a) controls the sequencing of one or more tasks within a logical unit; and
b) processes the task management functions ~~(see clause 7)~~(see clause 7).

4.5.22 Task set class

A task set class contains a task class ~~(see 4.5.23)~~(see 4.5.14).

Each instance of a task set class shall contain the following objects:

a) zero or more tasks.

The interactions among the tasks in a task set are determined by the requirements for task set management specified in ~~clause 8~~clause 8 and the ACA requirements specified in ~~5.8.1~~5.8.1. 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-3).

4.5.23 Task class

~~4.5.23.1 Task class overview~~

~~A task class represents the work associated with a command. There shall be one task class for each task that the device server has not started processing.~~

~~A task is represented by an I_T_L_Q nexus (see 4.7) and is composed of:~~

~~a) A definition of the work to be performed by the logical unit in the form of a command;~~
~~b) A task attribute (see 8.6) that allows the application client to specify processing relationships between various tasks in the task set; and~~
~~c) Optionally, a task priority (see 8.7).~~

~~4.5.23.2 Task tag attribute~~

~~A task tag attribute identifies a command. The I_T_L_Q nexus representing a task includes a task tag, allowing many uniquely identified tagged tasks to be present in a single task set. A task tag is composed of up to 64 bits.~~

~~A SCSI initiator device assigns task tag values for each I_T_L_Q nexus in a way that ensures that the nexus uniqueness requirements stated in this subclause are met. Transport protocols may define additional restrictions on task tag assignment (e.g., restricting task tag length, requiring task tags to be unique per I_T nexus or per I_T_L nexus, or sharing task tag values with other uses such as task management functions).~~

~~An I_T_L_Q 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 5.8.3). An I_T_L_Q nexus is unique if one or more of its components is unique within the specified time interval.~~

~~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 task tag.~~

For a description of the task class see 4.5.14.

4.5.24 Well known logical unit class

A well known logical unit class is a logical unit class (see 4.5.24.1) with the additional characteristics defined in this subclause.

Well known logical units are addressed using the well known logical unit addressing method (see 4.6.11) of extended logical unit addressing (see 4.6.10). Each well known logical unit has a well known logical unit number (W-LUN). W-LUN values are defined in SPC-3.

If a SCSI target port receives a W-LUN and the well known logical unit specified by the W-LUN does not exist, the task router shall follow the rules for selection of incorrect logical units described in ~~5.8.4~~5.8.4 and ~~7.12~~7.12.

If a well known logical unit is supported within a SCSI target device, then that logical unit shall support all the commands defined for it.

Access to well known logical units shall not be affected by access controls.

All well known logical units:

a) Shall not have logical unit names; and
b) Shall identify themselves using the SCSI target device names of the SCSI device in which they are contained.

NOTE 1 - A SCSI target device may have multiple SCSI target device names if the SCSI target device supports multiple SCSI transport protocols (see 4.5.19).

The name of the well known logical unit may be determined by issuing an INQUIRY command requesting the Device Identification VPD page (see SPC-3).