July 15, 1998

To: T10 Committee

From: Tom Coughlan
Compaq Computer Company
Mail Stop ZKO-3-4/U14
110 Spitbrook Road
Nashua, New Hampshire
Telephone: 603-884-0933
E-mail: tom.coughlan@digital.com

Subject: Changeable Device Identifiers

Summary

There is a requirement in some operating environments for the user to be able to associate a persistent identifier with a SCSI device. This proposal provides the requited functionality by adding two commands to SPC-2: Set Changeable Device Identifier and Report Changeable Device Identifier. These commands are defined for all device types. Implementation of these commands is optional.

Background

Many operating systems have traditionally assigned names to SCSI devices based on the path from the operating system (OS) to the device. These naming schemes are convenient because they provide names that are unique, and they convey some information about the logical or physical location of the device. Path-based names are problematic, however, when:

1. there are multiple paths from an OS instance to the device
2. there are multiple OS instances in the same naming domain with access to the same device (e.g. a cluster)
3. the interconnect employs path identifiers that are not as persistent as is required for OS device identification. Fibre Channel is one such interconnect.

One solution to these problems is to devise a device naming scheme that is based on Word Wide Identifiers (WWIDs). In such a scheme, each SCSI device is required to provide a persistent, path-independent, world-wide unique identifier in its Inquiry data. The operating system uses the WWID as the basis for device naming, thereby avoiding the problems with path-based naming listed above.

The WWID formats that are typically being implemented for SCSI are either 64-bit or 128-bit binary values. These identifiers are too long to be usable by humans as device identifiers. To resolve this difficulty, the OS (or a group of OS instances in a cluster) will typically assign a short alias to each WWID, and will present this to the user as the device

name. The OS is responsible for maintaining the alias-to-WWID mapping in such a way as to provide the device naming consistency and persistence that is expected by the users. For this discussion, we will assume that the OS implements a scheme for automatically assigning aliases to WWIDs, and that it implements a means for the user to modify aliases as desired to create meaningful device names.

The Problem

WWID-based naming does not work well in certain environments. For example:

1. In many system installations, the individuals who replace failed storage devices do not have access to the operating system. This presents a problem when WWID-based naming is in use, because the replacement storage device will, by default, have a different OS device name. The default OS name can be overridden, or OS parameters can be changed to compensate for the new name, but these actions require the involvement of personnel with different expertise, adding to the cost of the repair.
2. Producers of turn-key systems desire to ship and maintain identical copies of the OS on identically-configured hardware systems. This is not possible when WWID-based naming is in use, because the WWID-to-alias mapping on each system is necessarily different.
3. In some installations it is necessary to boot different instances of an operating system at different times. It is desirable for the device names in these environments to match. This is difficult to achieve in general with WWID-based naming, and may be impossible in an environment where there is a read-only (CD-ROM) system disk.
4. The default device name does not provide information about the location of the device. New tools are needed to provide the user with a mapping from the default WWID-based device name to the device's logical or physical location within the configuration. These tools may not be available in the same timeframe as when path-independent device names are required.

The Proposal

A variation on WWID-based naming is proposed to address the environments where the properties of WWID-based naming cause difficulties. In the proposed scheme, each SCSI device optionally implements a writable, persistent, device identifier.

Commands are provided to set and report a single fixed-length identifier. The command and response formats are designed to allow future extensions for multiple variable length identifiers, if the need arises. These extensions can be made in a manner that is similar to the format of the device identification page (83h) of the vital product data.

Detailed Proposal

Two commands are added to SPC-2, as follows:

REPORT CHANGEABLE DEVICE IDENTIFIER command

The REPORT CHANGEABLE DEVICE IDENTIFIER command (see table 1) requests that the logical unit send information about its changeable device identifier to the application client.

Table 1 --- REPORT CHANGEABLE DEVICE IDENTIFIER command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (tbd)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	Allocation length						
8								(LSB)
9	Control							

The REPORT CHANGEABLE DEVICE IDENTIFIER command shall not be affected by reservations or persistent reservations.

The Allocation length field in the command descriptor block indicates how much space has been reserved for the returned parameter data. If the length is not sufficient to contain all the parameter data, the first portion of the data shall be returned. This shall not be considered an error. The actual length of the parameter data is available in the Length field in the parameter data. If the remainder of the parameter data is required, the application client should send a new REPORT CHANGEABLE DEVICE IDENTIFIER command with an Allocation length field large enough to contain all the data.

REPORT CHANGEABLE DEVICE IDENTIFIER parameter data

The format for the parameter data provided in response to a REPORT CHANGEABLE DEVICE IDENTIFIER command is shown in table 2.

Table 2 --- REPORT CHANGEABLE DEVICE IDENTIFIER parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	Length (14h)							
1	Reserved							
2	Reserved							
3	Reserved							Valid
4	Reserved							
5	Reserved							
6	(MSB)							
	Identifier							
20								(LSB)

The Length field specifies the length in bytes of the parameter data. If the allocation length field of the command descriptor block is too small to transfer all of the identifier, the Length shall not be adjusted to reflect the truncation.

The Valid bit shall be clear until a SET CHANGEABLE DEVICE IDENTIFIER command with a non-zero length is successfully executed. The Valid bit shall remain set until a SET CHANGEABLE DEVICE IDENTIFIER command with a zero length is successfully executed.

The Identifier field contains the identifier that was last written by a successful SET CHANGEABLE DEVICE IDENTIFIER command with a non-zero length. The value of the Identifier field is not specified when the Valid bit is clear.

The value of the Identifier and the Valid bit shall be changed only by a SET CHANGEABLE DEVICE IDENTIFIER command. Their values persist through resets, power cycles, media format operations, and media replacement.

The logical unit shall return the same identification data to all initiators, on all of its ports.

The execution of a REPORT CHANGEABLE DEVICE IDENTIFIER command may require the enabling of 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 device server should complete the REPORT CHANGEABLE DEVICE IDENTIFIER command promptly, with CHECK CONDITION status if necessary, rather than wait for the device to become ready.

If a REPORT CHANGEABLE DEVICE IDENTIFIER command is received from an initiator with a pending unit attention condition (i.e., before the device server reports

CHECK CONDITION status), the device server shall perform the REPORT CHANGEABLE DEVICE IDENTIFIER command and shall not clear the unit attention condition (see SAM).

SET CHANGEABLE DEVICE IDENTIFIER command

The SET CHANGEABLE DEVICE IDENTIFIER command (see table 3) requests that the logical unit modify its changeable device identifier information.

Table 3 --- SET CHANGEABLE DEVICE IDENTIFIER command

Bit Byte	7	6	5	4	3	2	1	0
0	Operation code (tbd)							
1	Reserved							
2	Reserved							
3	Reserved							
4	Reserved							
5	Reserved							
6	Reserved							
7	(MSB)	Parameter list length = 0 or 10h						
8								(LSB)
9	Control							

The SET CHANGEABLE DEVICE IDENTIFIER command shall not be affected by reservations or persistent reservations.

The parameter list length field specifies the length in bytes of the parameter list that shall be transferred from the application client to the device server. The parameter list shall be 0 or 16 bytes in length and the Parameter list length field shall contain 0 or 16 (10h). If the Parameter list length is not 0 or 16, 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.

A parameter list length of zero indicates that no data shall be transferred, and the valid bit in subsequent REPORT CHANGEABLE DEVICE IDENTIFIER commands shall be clear.

If an application client sends a SET CHANGEABLE DEVICE IDENTIFIER command that changes the Identifier or the state of the Valid bit, the device server shall generate a unit attention condition for all initiators except the one that issued the SET CHANGEABLE DEVICE IDENTIFIER command (see SAM). The device server shall set the additional sense code to MODE PARAMETERS CHANGED.

SET CHANGEABLE DEVICE IDENTIFIER parameter data

The format for the parameter data provided with the SET CHANGEABLE DEVICE IDENTIFIER command is shown in table 4.

Table 4 --- SET CHANGEABLE DEVICE IDENTIFIER parameter data

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)							
				Identifier				
15								(LSB)

The Identifier field specifies the value of the changeable identifier that is to be returned in subsequent REPORT CHANGEABLE DEVICE IDENTIFIER commands.

Change to SAM-2

(The following changes are made with respect to SAM-2, Rev. 7.)

5.6.5 Unit Attention condition

A unit attention condition shall persist on the logical unit for each initiator until that initiator clears the condition as described in the following paragraphs.

If an INQUIRY command or a REPORT CHANGEABLE DEVICE IDENTIFIER command is received from an initiator to a logical unit with a pending unit attention condition (before the logical unit generates the auto contingent allegiance or contingent allegiance condition), the logical unit shall perform the INQUIRY or REPORT CHANGEABLE DEVICE IDENTIFIER command and shall not clear the unit attention condition.

If a request for sense data is received from an initiator with a pending unit attention condition (before the logical unit establishes the auto contingent allegiance or contingent allegiance condition), then the logical unit shall either:

a) Report any pending sense data and preserve the unit attention condition on the logical unit; or,
b) Report the unit attention condition.

If the second option is chosen (reporting the unit attention condition), the logical unit may discard any pending sense data and may clear the unit attention condition for that initiator.

If the logical unit has already generated the auto contingent allegiance or contingent allegiance condition for the unit attention condition, the logical unit shall perform the second action listed above. If NACA for the REQUEST SENSE command is zero and the command is untagged the contingent allegiance condition shall be cleared.

If an initiator issues a command other than INQUIRY, REPORT CHANGEABLE DEVICE IDENTIFIER, or REQUEST SENSE while a unit attention condition exists for that initiator (prior to generating the auto contingent allegiance or contingent allegiance condition for the unit attention condition), the logical unit shall not perform the command and shall report ACA ACTIVE (NACA=1, see 5.1.2) or BUSY (NACA=0) status.