February 2, 1998

To: T10 Committee

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

Subject: Writable Device Identifiers

Summary

There is a requirement in some environments for the user to be able to write an identifier to a SCSI device. This identifier is subsequently returned in the Device identification page of the Inquiry data. This functionality is provided by making two changes to the current standard, both in SPC-2: 1) define a new device identifier type value, to denote a user-written identifier, and 2) define a new mode page, to provide the application client with the means to set one or more device identifiers. Implementation of the writable identifier is optional, and the mode page is optional for all device types.

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. If the identifier is present, the SCSI device shall return it to the OS in the Device identification page of the Inquiry data. A new identifier type shall be assigned to designate the user-supplied identifier.

SCSI devices that have a management interface, such as RAID controllers, may provide a means for the user to specify a device identifier for each logical SCSI device provided by the controller. Once defined, the device shall include the identifier in the Device identification page.

I addition, a standard method is required to allow an identifier to be written to the SCSI device, particularly for devices that do not have a separate management interface. This is accomplished via a new mode page. The new mode page shall be optional, and shall be defined for all device types. For ease of implementation, the format of the new mode page resembles the format of the Device identification page. In typical applications, there would be just one identifier in the page, and the identifier would be short, because to the requirement for a human to be able to disambiguate it from others.

With these capabilities, it is possible to provide tools that can solve the problems described above.

Detailed Proposal

All the changes associated with this proposal are in SPC-2. The changes described in this document are based on Rev. 1 of SPC-2.

The first change is in Section 8.4.3, Table 114. The change defines a new identifier type for the writable identifier:

Table 114 — Identifier type

Value	Description
0h	No assignment authority was used and consequently there is no guarantee that the identifier is globally unique (i.e., the identifier is vendor-specific)
1h	The first 8 bytes of the identifier field are a Vendor ID (see annex C). The organization associated with the Vendor ID is responsible for ensuring that the remainder of the identifier field is unique. One recommended method of constructing the remainder of the identifier field is to concatenate the product identification field from the standard INQUIRY data field and the product serial number field from the unit serial number page.
2h	The identifier field contains an IEEE Extended Unique Identifier, 64-bit (EUI-64). In this case, the identifier length field shall be set to 8. Note that the IEEE guidelines for EUI-64 specify a method for unambiguously encapsulating an IEEE 48-bit identifier within an EUI-64.
3h	The identifier field contains an FC-PH Name_Identifier. Any FC-PH identifier may be used, including one of the four based on the IEEE company_id.
4h	The identifier field contains an identifier that was provided by the user. There is no guarantee that the identifier is globally unique.
5h- Fh	Reserved

The second and final change is the definition of a new mode page, to be added to Section 8.3.

8.3.n Device identification mode page

The Device identification mode page (see table n) provides the application client the means to set one or more device identifiers. These identifiers shall subsequently be returned by the logical unit in the Device identification page of the Inquiry data. (Identifiers that are created through this mechanism shall have an Identifier type equal to four.)

Table --- n

Bit Byte	7	6	5	4	3	2	1	0
0	PS	Reserved	Page code (TBD)					
1	Page length (n-1)							
	Identification descriptor list							
2	Identification descriptor (first)							
	.							
	.							
	.							
	Identification descriptor (last)							
N								

The parameters savable (PS) bit is only used with the MODE SENSE command. This bit is reserved with the MODE SELECT command. A PS bit of one indicates that the device server is capable of saving the page in a non-volatile vendor-specific location. If the PS bit is one in MODE SENSE data then the page shall be savable by issuing a MODE SELECT command with the SP bit of one.

Each Identification descriptor (see Table m) contains information identifying the logical unit. If the Logical Unit is accessible through any other path, it shall return the same identification.

Table m ---Identification descriptor

Bit Byte	7	6	5	4	3	2	1	0
0	Reserved				Code set			
1	Reserved							
2	Reserved							
3	Identifier length (n-3)							
4	(MSB)							
	Identifier							
n								(LSB)

The Code set, Identifier length and Identifier fields in table m are as defined in “8.4.3 Device identification page”.

The MODE SENSE data shall contain all the user supplied Identifiers (Identifier type equal to four) that the Logical Unit provides in the Device identification page of the Inquiry data. If there are no Identifiers with an Identifier type equal to four, then the Page length returned in the MODE SENSE shall have a value of zero.

A MODE SELECT command shall cause all the user supplied Identifiers (Identifier type equal to four) that the Logical Unit provides in the Device identification page of the Inquiry data to be replaced by the Identification descriptor list contained in the MODE SELECT data. A MODE SELECT command with Page length equal to zero shall cause all existing user supplied Identifiers to be removed from the Logical Unit.