

To: T10 Technical Committee  
From: Rob Elliott, HP (elliott@hp.com)  
Date: 7 September 2006  
Subject: 06-221r2 SPC-4 Peripheral device identifying information

### **Revision history**

Revision 0 (3 May 2006) First revision

Revision 1 (10 June 2006) Incorporated comments from May 2006 CAP WG - changed to SET/REPORT DEVICE IDENTIFIER commands instead of the Device Identification VPD page 83h.

Revision 2 (7 September 2006) Incorporated comments from July 2006 CAP WG - renamed to SET/REPORT IDENTIFYING INFORMATION to avoid overuse of "identifier", increased the information type field to 7 bits, added a descriptor list indicating the types supported and the maximum length of each. Added a short model section.

### **Related documents**

spc4r04 - SCSI Primary Commands - 4 (SPC-4) revision 4

06-278 - SPC-4 Fix persistence of SET DEVICE IDENTIFIER (Paul Entzel, Quantum)

### **Overview**

Worldwide unique logical unit names in 64-bit and 128-bit binary NAA and EUI-64-based formats or (up to) 256 byte UTF-8 SCSI name string formats are great for applications, but are unwieldy for users.

This proposal defines a new UTF-8 string format identifier that can be used to provide a descriptive string about the logical unit. The intent is that it be used for display purposes in management applications. UTF-8 rather than ASCII allows characters from other languages to be used if appropriate.

Examples:

- a) "Database log storage"
- b) "Database table storage"
- c) "Server 34 boot drive"

Unlike the NAA, EUI-64, and SCSI name string identifiers in the Device Identification VPD page 83h, there is no requirement that it be worldwide unique (or even unique within the target device).

The text identifier is read with the REPORT DEVICE IDENTIFIER command (renamed to REPORT IDENTIFYING INFORMATION) and may be written with the SET DEVICE IDENTIFIER command (renamed to SET IDENTIFYING INFORMATION) or with a vendor-specific mechanism.

### **Suggested changes**

**3.1.31 device type:** The type of peripheral device (i.e., device model) implemented by the device server and indicated by the contents of the PERIPHERAL DEVICE TYPE field in the standard INQUIRY data (see 6.4.2).

**3.1.xx peripheral device:** [An object that that connects to a logical unit.](#)

**3.1.xx peripheral device identifying information:** [A value associated with a peripheral device accessed via the REPORT IDENTIFYING INFORMATION command \(see 6.20\) and SET IDENTIFYING INFORMATION command \(see 6.31\). See 5.xx.](#)

**3.1.xx peripheral device text identifying information:** [A UTF-8 format string associated with a peripheral device accessed via the REPORT IDENTIFYING INFORMATION command \(see 6.20\) and SET IDENTIFYING INFORMATION command \(see 6.31\). See 5.xx.](#)

**3.1.xx peripheral device type:** [Synonym for device type \(see 3.1.31\).](#)

## **5 Model common to all device types**

### **5.6 Reservations**

---

---

Editor's Note 1: Change REPORT/SET DEVICE IDENTIFIER to REPORT/SET IDENTIFYING

---



---

 INFORMATION in the reservation table
 

---



---

**5.xx Identifying information**

The REPORT IDENTIFYING INFORMATION command (see 6.20) and the SET IDENTIFYING INFORMATION command (see 6.31) allow an application client to maintain one or more sets of identifying information associated with the peripheral device.

Identifying information shall persist through power cycles (i.e., be stored in non-volatile storage), hard resets, logical unit resets, I T nexus losses, media format operations, and media replacement.

Table 1 defines the identifying information types.

**Table 1 — Identifying information types**

<b>Code</b>	<b>Identifying information</b>	<b>Length</b>	<b>Format of parameter data</b>
<a href="#">0000000b</a>	<a href="#">Peripheral device identifying information - a value describing the peripheral device (e.g., an operating system volume label).</a>	<a href="#">0 through 64 bytes required; 65 through 512 bytes optional</a>	<a href="#">6.20.2</a>
<a href="#">0000010b</a>	<a href="#">Peripheral device text identifying information - a null-terminated (see 4.4.2) UTF-8 format string providing an informational description of the peripheral device (e.g., a descriptive string entered by a system administrator).</a>	<a href="#">0 through 256 bytes</a>	<a href="#">6.20.2</a>
<a href="#">xxxxxx1b</a> <sup>a</sup>	<a href="#">Restricted for SCC-2.</a>		
<a href="#">All others</a>	<a href="#">Reserved</a>		
<sup>a</sup> <a href="#">x represents a bit set to either 0 or 1.</a>			

Identifying information is changed by:

- a) [the SET IDENTIFYING INFORMATION command \(see 6.31\); or](#)
- b) [a mechanism outside the scope of this standard \(e.g., a system administrator may be able to change identifying information through a management interface\).](#)

**6 Commands for all device types****6.1 Summary of commands for all device types**


---



---

 Editor's Note 2: Change REPORT/SET DEVICE IDENTIFIER to REPORT/SET IDENTIFYING INFORMATION in the command table and in the annexes
 

---



---

**6.20 REPORT ~~DEVICE IDENTIFIER~~ IDENTIFYING INFORMATION command****6.20.1 REPORT IDENTIFYING INFORMATION command overview**

The REPORT ~~DEVICE IDENTIFIER~~ IDENTIFYING INFORMATION command (see table 146) requests that the device server send ~~device identification information~~ identifying information (see 5.xx) to the application client. ~~As defined in the SCC-2 standard, t~~The REPORT ~~DEVICE IDENTIFIER~~ IDENTIFYING INFORMATION command is [an extension to](#) the REPORT PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action of the MAINTENANCE IN command [defined in SCC-2](#). Additional MAINTENANCE IN and MAINTENANCE OUT service actions are defined in SCC-2 and in this standard.

The MAINTENANCE IN service actions defined only in SCC-2 shall apply only to ~~SGSI devices~~ logical units that return a device type of 0Ch (i.e., storage array controller device) or the sccs bit set to one in their standard INQUIRY data (see 6.4.2). When a ~~SGSI device~~ logical unit returns a device type of 0Ch or the sccs bit set to one in its standard INQUIRY data, the implementation requirements for the SCC-2 MAINTENANCE IN service actions shall be as specified in SCC-2. Otherwise the MAINTENANCE IN service action definitions and implementation requirements stated in this standard shall apply.

~~The logical unit shall return the same identifier to all application clients.~~

The device server shall return the same identifying information regardless of the I\_T nexus being used to retrieve the identifying information.

Processing a REPORT ~~DEVICE IDENTIFIER~~ IDENTIFYING INFORMATION may require the enabling of a nonvolatile memory within the logical unit. If the nonvolatile memory is not ready, the command shall be terminated with CHECK CONDITION status, rather than wait for the nonvolatile memory to become ready. The sense key shall be set to NOT READY and the additional sense code shall be set as described in table 193 (see 6.35). This information should allow the application client to determine the action required to cause the device server to become ready.

**Table 146 — REPORT ~~DEVICE IDENTIFIER~~ IDENTIFYING INFORMATION command**

Byte\Bit	7	6	5	4	3	2	1	0	
0	OPERATION CODE (A3h)								
1	Reserved			SERVICE ACTION (05h)					
2	Reserved								
3	Reserved								
4	Restricted <u>for SCC-2</u>								
5	Reserved								
6	(MSB)	ALLOCATION LENGTH						(LSB)	
9	Reserved								
10	Reserved						Restricted	Reserved	
	<u>INFORMATION TYPE</u>							Reserved	
11	CONTROL								

~~SCC-2 defines specific usages for bytes 4 and 5, and bit 1 in byte 10, however these fields are reserved for the REPORT DEVICE IDENTIFIER command defined by this standard.~~

The ALLOCATION LENGTH field is defined in 4.3.4.6.

The INFORMATION TYPE field is defined in table 147.

**Table 147 — INFORMATION TYPE field**

<u>Code</u>	<u>Description of parameter data</u>	<u>Format of parameter data</u>
<a href="#">0000000b</a>	<a href="#">Peripheral device identifying information (see table 1 in 5.xx).</a>	<a href="#">6.20.2</a>
<a href="#">0000010b</a>	<a href="#">Peripheral device text identifying information (see table 1 in 5.xx).</a>	
<a href="#">111110b</a>	<a href="#">Identifying information supported.</a> <a href="#">The parameter data contains a list of supported identifying information types and the maximum length of each.</a>	<a href="#">6.20.3</a>
<a href="#">xxxxxx1b</a> <sup>a</sup>	<a href="#">Restricted for SCC-2.</a>	
<a href="#">All others</a>	<a href="#">Reserved</a>	
<sup>a</sup> x represents a bit set to either 0 or 1.		

**[6.20.2 IDENTIFYING INFORMATION parameter data](#)**

~~The REPORT\_DEVICE\_IDENTIFIER\_parameter data (see table 148) contains a four-byte field that contains the length in bytes of the parameter data and the logical unit's identifier.~~

[Table 148 defines the IDENTIFYING INFORMATION parameter data.](#)

**Table 148 — ~~REPORT\_DEVICE\_IDENTIFIER~~IDENTIFYING INFORMATION parameter data**

Byte\Bit	7	6	5	4	3	2	1	0	
0	(MSB)	<del>IDENTIFIER</del> INFORMATION LENGTH (n-3)							
3								(LSB)	
4		<del>IDENTIFIER</del> INFORMATION							
n									

The ~~IDENTIFIER~~INFORMATION LENGTH field indicates the length in bytes of the ~~IDENTIFIER~~INFORMATION field. The relationship between the ~~IDENTIFIER~~INFORMATION LENGTH field and the CDB ALLOCATION LENGTH field is defined in 4.3.4.6. ~~The identifier length shall initially equal zero, and shall be changed only by a successful SET\_DEVICE\_IDENTIFIER\_command.~~

~~The IDENTIFIER field shall contain a vendor specific value. The value reported shall be the last value written by a successful SET\_DEVICE\_IDENTIFIER\_command. The value of the identifier shall be changed only by a SET\_DEVICE\_IDENTIFIER\_command. The identifier value shall persist through logical unit resets, I\_T nexus losses, media format operations, and media replacement.~~

[The INFORMATION field contains the identifying information with the specified identifying information type.](#)

**[6.20.3 IDENTIFYING INFORMATION SUPPORTED parameter data](#)**

Table 149 defines the IDENTIFYING INFORMATION SUPPORTED parameter data.

**Table 149 — IDENTIFYING INFORMATION SUPPORTED parameter data**

Byte\Bit	7	6	5	4	3	2	1	0
0	<u>(MSB)</u>							
	<u>IDENTIFYING INFORMATION LENGTH (n-3)</u>							
3	<u>(LSB)</u>							
<u>Identifying information descriptor list</u>								
4	<u>Identifying information descriptor (first)(see table 150)</u>							
7								
	...							
n - 3	<u>Identifying information descriptor (last)(see table 150)</u>							
n								

The IDENTIFYING INFORMATION LENGTH field indicates the length in bytes of the identifying information descriptor list. The relationship between the IDENTIFYING INFORMATION DESCRIPTOR LIST LENGTH field and the CDB ALLOCATION LENGTH field is defined in 4.3.4.6.

The identifying information descriptor list contains an identifying information descriptor (see table 150) for each identifying information supported by the device server. The identifying information descriptors shall be sorted in incrementing order by information type.

Table 150 defines the identifying information descriptor.

**Table 150 — Identifying information descriptor**

Byte\Bit	7	6	5	4	3	2	1	0
0	<u>INFORMATION TYPE</u>							
1	<u>(MSB)</u>							
	<u>MAXIMUM INFORMATION LENGTH</u>							
3	<u>(LSB)</u>							

The INFORMATION TYPE field indicates the information type.

The MAXIMUM INFORMATION LENGTH field indicates the maximum length of the identifying information for the indicated information type, in bytes.

### 6.31 SET ~~DEVICE IDENTIFIER~~IDENTIFYING INFORMATION command

#### 6.31.1 SET IDENTIFYING INFORMATION command overview

The SET ~~DEVICE IDENTIFIER~~IDENTIFYING INFORMATION command (see table 181) requests that the device ~~server identifier information in the logical unit be~~ set ~~identifying information (see 5.xx) in the logical unit~~ to the value received in the SET ~~DEVICE IDENTIFIER~~IDENTIFYING INFORMATION parameter list. ~~As defined in the SCC-2 standard, t~~The SET ~~DEVICE IDENTIFIER~~IDENTIFYING INFORMATION command is ~~an extension to~~ the SET PERIPHERAL DEVICE/COMPONENT DEVICE IDENTIFIER service action of the MAINTENANCE OUT command ~~defined in SCC-2~~. Additional MAINTENANCE IN and MAINTENANCE OUT service actions are defined in SCC-2 and in this standard.

The MAINTENANCE OUT service actions defined only in SCC-2 shall apply only to ~~SCSI devices~~logical units that return a device type of 0Ch (~~i.e., storage array controller device~~) or the sccs bit set to one in their standard INQUIRY data (~~see 6.4.2~~). When a ~~SCSI devices~~logical unit returns a device type of 0Ch or the sccs bit set to one in its standard INQUIRY data, the implementation requirements for the SCC-2 MAINTENANCE OUT service actions shall be as specified in SCC-2. Otherwise the MAINTENANCE OUT service action definitions and implementation requirements stated in this standard shall apply.

On successful completion of a SET ~~DEVICE IDENTIFIER~~IDENTIFYING INFORMATION command that changes ~~the device identifier~~identifying information saved by the logical unit, the device server shall establish a unit attention condition (see SAM-3) for the initiator port associated with every I\_T nexus except the I\_T nexus on which the SET ~~IDENTIFIER~~IDENTIFYING INFORMATION command was received, with the additional sense code set to DEVICE IDENTIFIER CHANGED.

Table 181 — SET ~~DEVICE IDENTIFIER~~IDENTIFYING INFORMATION command

Byte\Bit	7	6	5	4	3	2	1	0	
0	OPERATION CODE (A4h)								
1	Reserved			SERVICE ACTION (06h)					
2	Reserved								
3	Reserved								
4	Restricted for SCC-2								
5	Reserved								
6	(MSB)	PARAMETER LIST LENGTH						(LSB)	
9	Reserved								
10	Reserved					Restricted		Reserved	
	INFORMATION TYPE								
11	CONTROL								

~~SCC-2 defines specific usages for bytes 4 and 5, and bit 1 in byte 10, however these fields are reserved for the SET DEVICE IDENTIFIER command defined by this standard.~~

The PARAMETER LIST LENGTH field specifies the length in bytes of the ~~identifier~~identifying information that shall be transferred from the application client to the device server. ~~The maximum value for this field shall be 512 bytes. A parameter list length of zero specifies that no data shall be transferred, and that subsequent REPORT DEVICE IDENTIFIER commands shall return an Identifier length of zero. Logical units that implement this command shall be capable of accepting a parameter list length of 64 bytes or less. If the parameter list length exceeds 64 bytes and the logical unit is not capable of storing the requested number of bytes, then the command shall be terminated with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.~~

The INFORMATION TYPE field specifies the identifying information type and is defined in table 182.

**Table 182 — INFORMATION TYPE field**

<u>Code</u>	<u>Definition of the parameter data</u>
<u>0000000b</u>	<p><u>Peripheral device identifying information (see table 1 in 5.xx).</u></p> <p>If the PARAMETER LIST LENGTH field is set to greater than the maximum length of the peripheral device identifying information, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p>
<u>0000010b</u>	<p><u>Peripheral device text identifying information (see table 1 in 5.xx).</u></p> <p>If the PARAMETER LIST LENGTH field is set to a value greater than the maximum length of the peripheral device text identifying information (i.e., 256), the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN CDB.</p> <p>If the format of the INFORMATION field is incorrect, the device server shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to INVALID FIELD IN PARAMETER LIST.</p>
<u>xxxxxx1b<sup>a</sup></u>	<u>Restricted for SCC-2.</u>
<u>All others</u>	<u>Reserved</u>
<sup>a</sup> x represents a bit set to either 0 or 1.	

**6.31.2 SET IDENTIFYING INFORMATION parameter list**

The SET DEVICE IDENTIFIER parameter list (see table 183) contains the identifier to be set by the addressed logical unit.

Table 183 defines the SET IDENTIFYING INFORMATION parameter list.

**Table 183 — SET ~~DEVICE IDENTIFIER~~ IDENTIFYING INFORMATION parameter list**

<u>Byte\Bit</u>	<u>7</u>	<u>6</u>	<u>5</u>	<u>4</u>	<u>3</u>	<u>2</u>	<u>1</u>	<u>0</u>
<u>0</u>	<del>IDENTIFIER</del> INFORMATION							
<u>n</u>								

The IDENTIFIER field is a value selected by the application client using mechanisms outside the scope of this standard to be returned in subsequent REPORT DEVICE IDENTIFIER commands.

The INFORMATION field specifies the identifying information for the specified information type (see table 182 in 6.31.1).