

From: Gerry Houlder, Seagate Technology <gerry.houlder@seagate.com>
Subj: SPC-4 Read debug log proposal
Date: Jan. 16, 2007

This document proposes a standard method to control and retrieve debug data related to device errors. Several drive vendors (including Seagate) have proprietary methods for controlling these operations today, but customers have asked that a standard method be defined so their processes can be simpler.

Black text indicates current wording in the SPC-4 rev. 5a draft standard. Blue underlined text is new.

Rev. 1 changes requested at 9/13 meeting:

1. Rename "application log" as "debug log";
2. Added resumption of debug log updates when resets occur;
3. Add statement after resume that "debug log may include data captured while log update was suspended";
4. Added CLR_SUP bit in READ BUFFER Retrieve Debug Log data;
5. In table new2, increase Maximum Available Length to 4 bytes;
6. Add statement to Write Buffer defining action when debug log update is suspended;

Rev. 2 has editorial changes requested at 11/8/06 CAP meeting in 5.11.3.

Rev. 3 has changes requested at 1/16/07 CAP meeting.

1. Addition to model section of SPC-4:

5.11 Setting and retrieving error history

5.11.1 Error history overview

The optional error history is vendor specific data that has been collected by a SCSI device to aid in troubleshooting device errors. The WRITE BUFFER command (see 6.36.14) provides a method of inserting application client error history information into the SCSI device error history or clearing the error history. The READ BUFFER command (see 6.14.9) provides a method of retrieving error history from the SCSI device.

5.11.2 Application client logging

Application client logging is a method the application client may use to store application client detected error information in a logical unit's non-volatile storage (see 6.36.14). The information the application client sends to the logical unit is appended to the error history. The application client error information ~~may be recovered as part of the SCSI device error history (see 5.11.3) or~~ by means outside the scope of this standard and is not used for any logical unit related error recovery.

Deleted: is
Formatted
Formatted

A log that contains a mix of application client error information and logical unit error information may be used to correlate an application client error with any errors internal to the logical unit. This ~~provides a vendor independent way of correlating error logs.~~

Deleted: does not replace the vendor specific methods for collecting and analyzing engineering data, but

Application clients should minimize the amount of error information that is requested to be logged to prevent log overflows.

5.11.3 Retrieving error history

Device servers may allow the error history to be retrieved using a sequence of READ BUFFER commands. If the required command sequence is not received, a READ BUFFER command that is received in the incorrect sequence shall be terminated with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to COMMAND SEQUENCE ERROR.

All device servers that support retrieving error history using READ BUFFER commands shall support the following sequence:

- 1) Transfer the table of entries that defines the available error history buffers in response to a READ BUFFER command with the MODE field set to retrieve error history (i.e., 1Ch), the BUFFER ID field set to 00h, the BUFFER OFFSET field set to zero, and the ALLOCATION LENGTH field set large enough to transfer the entire table of entries. As part of processing this READ BUFFER command the device server shall suspend updating the error history data;
- 2) Retrieve the buffers in any order in response to a READ BUFFER command with the BUFFER ID value set to any supported value between 01h and FEh. A buffer may be retrieved using a single READ BUFFER command or multiple READ BUFFER commands. Device servers shall support the retrieval of error history using multiple READ BUFFER commands in the following sequence:
 - 1) Return error history data in response to the initial READ BUFFER command with the BUFFER ID field set to the desired buffer value, the BUFFER OFFSET field set to zero, and the ALLOCATION LENGTH field set to the desired transfer size;
 - 2) After GOOD status has been returned for the previous READ BUFFER command and the number of bytes transferred equals the allocation length, return error history data in response to another READ BUFFER command with BUFFER ID field set to the previous buffer value, the BUFFER OFFSET field set to the previous buffer offset plus the previous allocation length value, and the ALLOCATION LENGTH field set to the previous allocation length value;
 - 3) Repetition of step 2) until the number of bytes transferred is less than the allocation length value or a command sequence error is returned.
- 3) Repetition of step 2) until all of the required buffers have been transferred.

The device server shall resume updating error history when:

- a) the application client issues a READ BUFFER command with the MODE field set to retrieve error history, the BUFFER ID field set to FFh, the BUFFER OFFSET field set to zero, and the ALLOCATION LENGTH set to zero;
- b) a vendor specific timer expires;
- c) a power on occurs;
- d) a hard reset occurs;
- e) an I_T nexus loss occurs; or
- f) a logical unit reset occurs.

When updating of error history is resumed, the error history may include data captured while error history updating was suspended.

Error history may also be retrieved by vendor specific methods or other READ BUFFER command sequences that are outside the scope of this standard.

5.11.4 Interpreting the error history

The error history is interpreted using a vendor specific parsing application. Any other use of the error history is undefined.

[5.11.5 Clearing the error history](#)

The error history may be cleared by issuing a WRITE BUFFER command (see 6.36.14) with the BUFFER ID field set to download error history (i.e., 1Ch), the BUFFER OFFSET field set to zero, the parameter list length set to 00001Ah, and parameter data with the CLR bit set and the other fields set to zero.

2. Additions to READ BUFFER command

6.14.1 READ BUFFER command introduction

The READ BUFFER command (see table 126) is used in conjunction with the WRITE BUFFER command as a diagnostic function for testing memory in the SCSI device and the integrity of the service delivery subsystem. This command shall not alter the medium.

Table 126 – READ BUFFER command

Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Ch)							
1	RESERVED				MODE			
2	BUFFER ID							
3	(MSB) BUFFER OFFSET (LSB)							
5	(LSB)							
6	(MSB) ALLOCATION LENGTH (LSB)							
8	(LSB)							
9	CONTROL							

The function of this command and the meaning of fields within the CDB depend on the contents of the MODE field. The MODE field is defined in table 127.

Table 127 – READ BUFFER MODE field

MODE	Description	Reference
00h	Combined header and data ^a	6.14.2
01h	Vendor Specific ^a	6.14.3
02h	Data	6.14.4
03h	Descriptor	6.14.5
0Ah	Echo buffer	6.14.6
0Bh	Echo buffer descriptor	6.14.7
1Ah	Enable expander communications protocol and echo buffer	6.14.8
1Ch	Retrieve error history	5.11 and 6.14.9
04h - 09h	Reserved	
0Ch - 19h	Reserved	
1Bh	Reserved	
1Dh - 1Fh	Reserved	

^a Modes 00h and 01h are not recommended.

[Editors Note: clauses 6.14.2 through 6.14.8 are unchanged]

6.14.9 Retrieve error history mode (1Ch)

6.14.9.1 Retrieve error history overview

In this mode, the Data-In Buffer is filled with either a table of entries that describes the supported buffers (see 6.14.9.2) or vendor specific error history data (see 6.14.9.3). The BUFFER ID field specifies which part of the error history data shall be transferred. If the BUFFER ID field is set to an unsupported buffer, 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 BUFFER OFFSET field contains the byte offset within the specified buffer from which data shall be transferred. The application client should conform to the offset boundary requirements returned in the READ BUFFER descriptor (see 6.14.5). If the device server is unable to accept the specified buffer offset, 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.

6.14.9.2 Table of entries data description

When the BUFFER ID field is set to zero, a table of entries shall be transferred (table new1).

Table new1 – Table of entries format

Bit	7	6	5	4	3	2	1	0	
Byte 0	T10 VENDOR IDENTIFICATION								
7									
8	VERSION								
9	RESERVED							CLR_SUP	
10	RESERVED								
11	RESERVED								
12	RESERVED								
13	RESERVED								
14	(MSB)	DATA LENGTH (n-15)						(LSB)	
15	First table entry								
16									
23	:								
n-7	Last table entry								
n									

The T10 VENDOR IDENTIFICATION field contains eight bytes of left-aligned ASCII data (see 4.4.1) identifying the manufacturer of the product. The T10 vendor identification shall be one assigned by INCITS. A list of assigned T10 vendor identifications is in Annex E and on the T10 web site (<http://www.T10.org>).

The VERSION field indicates a vendor specific version.

The Clear Support (CLR_SUP) bit set to one indicates that the CLR function (see 6.36.14) is supported. The CLR_SUP bit set to zero indicates that the CLR function is not supported.

The DATA LENGTH field indicates the number of table entry bytes available to be transferred. This value shall not be altered even if the allocation length is not sufficient to transfer all of the available bytes.

Each table entry shall be 8 bytes as defined in table new2. There shall be an entry for each supported buffer ID. The first entry shall be for buffer ID zero and the entries shall be in order of ascending buffer IDs. The supported buffer IDs are not required to be contiguous.

There shall not be an entry for buffer ID of FFh because there shall be no data associated with this value.

Table new2 – Table entry format

Bit	7	6	5	4	3	2	1	0
0	SUPPORTED BUFFER ID							
1	RESERVED							
2	RESERVED							
3								
4	(MSB)							
5	MAXIMUM AVAILABLE LENGTH							
6								
7	(LSB)							

The SUPPORTED BUFFER ID field indicates the error history buffer ID associated with this table entry.

The MAXIMUM AVAILABLE LENGTH field indicates the maximum number of data bytes contained in this buffer. The actual number of bytes available for transfer may be smaller. For buffer ID 00h, this is the length of the table of entries.

6.14.9.3 Error history data description

When the BUFFER ID field is from 01h to FEh, vendor specific error history data is transferred.

When the BUFFER ID field is FFh, no data shall be available to be transferred.

3.0 Additions to WRITE BUFFER command

6.36.1 WRITE BUFFER command introduction

The WRITE BUFFER command (see table 195) is used in conjunction with the READ BUFFER command as a diagnostic function for testing logical unit memory in the SCSI target device and the integrity of the service delivery subsystem. Additional modes are provided for:

- a) Downloading microcode;
- b) Downloading and saving microcode;

- c) Downloading microcode with deferred activation; and
- d) Downloading [error history](#) (see 5.11).

Table 195 – WRITE BUFFER command

Bit	7	6	5	4	3	2	1	0
0	OPERATION CODE (3Bh)							
1	RESERVED			MODE				
2	BUFFER ID							
3	(MSB)	BUFFER OFFSET						(LSB)
5	(LSB)							
6	(MSB)	PARAMETER LIST LENGTH						(LSB)
8	(LSB)							
9	CONTROL							

The command shall not alter any medium of the logical unit when the data mode or the combined header and data mode is specified.

The function of this command and the meaning of fields within the CDB depend on the contents of the MODE field. The MODE field is defined in table 196.

Table 196 – WRITE BUFFER MODE field

MODE	Description	Reference
00h	Combined header and data ^a	6.36.2
01h	Vendor Specific ^a	6.36.3
02h	Data	6.36.4
04h	Download microcode	6.36.5
05h	Download microcode and save	6.36.6
06h	Download microcode with offsets ^b	6.36.7
07h	Download microcode with offsets and save ^b	6.36.8
0Ah	Echo buffer	6.36.9
0Eh	Download microcode with offsets and defer activation ^b	6.36.10
0Fh	Activate deferred microcode	6.36.11
1Ah	Enable expander communications protocol and echo buffer	6.36.12
1Bh	Disable expander communications protocol	6.36.13
1Ch	Download error history	5.11 and 6.36.14
03h	Reserved	
08h - 09h	Reserved	
0Bh - 0Dh	Reserved	
10h - 19h	Reserved	
1Dh - 1Fh	Reserved	

^aModes 00h and 01h are not recommended.
^bWhen downloading microcode with buffer offsets, the WRITE BUFFER command mode should be 06h, 07h, or 0Eh.

[Editors Note: clauses 6.36.2 through 6.36.13 are unchanged]

6.36.14 Download [error history](#) mode (1Ch)

In this mode the device server transfers data from the application client and stores it in an [error history buffer](#) (see 5.11). The format of the [error history](#) data is as specified in table 197. The BUFFER ID field and BUFFER OFFSET field are ignored in this mode.

Upon successful completion of a WRITE BUFFER command the data shall be appended to the [error history](#). [The error history updates requested in a WRITE BUFFER command are appended to the error history even if error history updating is suspended \(see 5.11.3\).](#)

The PARAMETER LIST LENGTH field specifies the maximum number of bytes that shall be transferred from the Data-Out Buffer to be stored in the [error history](#). If the PARAMETER LIST LENGTH field specifies a transfer that exceeds the [error history buffer](#)'s capacity, 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.

Table 197 – [Error history](#) data WRITE BUFFER format

Bit	7	6	5	4	3	2	1	0	
Byte									
0	(MSB)	T10 VENDOR IDENTIFICATION							(LSB)
7									
8	(MSB)	ERROR TYPE							(LSB)
9									
10								CLR	
11									
12	(MSB)	TIME STAMP							(LSB)
17									
18									
19									
20			RESERVED				CODE SET		
21									
22	(MSB)	ERROR LOCATION LENGTH (m-25)							(LSB)
23									
24	(MSB)	VENDOR SPECIFIC LENGTH (n-m)							(LSB)
25									
26	(MSB)	ERROR LOCATION							(LSB)
m									
m+1									
n									

The T10 VENDOR IDENTIFICATION field contains eight bytes of left-aligned ASCII data (see 4.4.1) identifying the vendor of the product. The T10 vendor identification shall be one assigned by INCITS. A list of assigned T10 vendor identifications is in Annex E and on the T10 web site (<http://www.T10.org>).

The ERROR TYPE field (see table 198) specifies the error detected by the application client.

Table 198 – ERROR TYPE field

CODE	Description
0000h	No error specified by the application client
0001h	An unknown error was detected by the application client
0002h	The application client detected corrupted data
0003h	The application client detected a permanent error
0004h	The application client detected a service response of SERVICE DELIVERY OR TARGET FAILURE (SAM-3).
0005h – 7FFFh	Reserved
8000h – FFFFh	Vendor specific

[A CLR bit set to one specifies that the error history shall be cleared except for error history contents that are not allowed to be cleared by the application client. The other fields \(see table 197\) shall be ignored. A CLR bit set to zero specifies that the debug buffer contents shall be preserved.](#)

The TIME STAMP field shall contain:

- a) The number of milliseconds that have elapsed since midnight, 1 January 1970 UT (see 3.1.124); or
- b) Zero, if the application client is not able to determine the UT of the log entry.

The CODE SET field (see table 199) specifies the code set used for the [error history](#) information and shall only apply to information contained in the VENDOR SPECIFIC field.

NOTE 33 - The CODE SET field is intended to be an aid to software that displays the [error history](#) information.

Table 199 – CODE SET field

Code	Description
0h	Reserved
1h	The error history information is binary
2h	The error history information is ASCII printable characters (i.e., code values 20h through 7Eh)
3h	The error history information is ISO/IEC 10646-1 (UTF-8) codes
4h - Fh	Reserved

The ERROR LOCATION FORMAT field (see table 200) specifies the format of the ERROR LOCATION field.

Table 200 – ERROR LOCATION FORMAT field

Code	Description
00h	No error specified by the application client
01h	The error location field specifies the logical block (e.g., LBA) associated with the error information contained within the error history .
02h – 7Fh	Reserved
80h - FFh	Vendor specific

The ERROR LOCATION LENGTH field specifies the length of the ERROR LOCATION field. The ERROR LOCATION LENGTH field value shall be a multiple of four. An error location length value of zero specifies there is no error location information.

The VENDOR SPECIFIC LENGTH field specifies the length of the VENDOR SPECIFIC field. The VENDOR SPECIFIC LENGTH field value shall be a multiple of four. A vendor specific length value of zero specifies there is no vendor specific information.

The ERROR LOCATION field specifies the location at which the application client detected the error.

The VENDOR SPECIFIC field provides vendor specific information on the error.