To: T10 Technical Committee
From: Gerry Houlder, Seagate Technology (gerry.houlder@seagate.com) and
Rob Elliott, HP (elliott@hp.com)
Date: 28 February 2007
Subject: 06-362r4 SPC-4 Error history

Revision history

Revision 0 (28 July 2006) First revision

Revision 1 (7 November 2006) Incorporated changes requested at 9/13/2006 CAP WG meeting:

a) Rename "application log" to "debug log";
b) Added resumption of debug log updates when resets occur;
c) Add statement after resume that "debug log may include data captured while log update was suspended";
d) Added CLR_SUP bit in READ BUFFER Retrieve Debug Log data;
e) In table new2, increase MAXIMUM AVAILABLE LENGTH field to 4 bytes; and
f) Add statement to WRITE BUFFER defining action when debug log update is suspended;

Revision 2 (15 January 2007) Incorporated editorial changes in 5.11.3 requested at 11/8/2006 CAP WG meeting.

Revision 3 (16 January 2007) Incorporated changes requested at 1/16/2007 CAP WG meeting.

Revision 4 (28 February 2007) Incorporated changes from Rob Elliott, HP. Includes: In READ BUFFER mode 1Ch, changed “table of entries” to “error history directory” and “table entry” to “error history.” In WRITE BUFFER mode 1Ch, changed “vendor specific” field to “application client error history.” Reworded everything in terms of taking an error history snapshot and reading from it. Added a unit attention condition called ERROR HISTORY SNAPSHOT RELEASED when the vendor-specific timer expires and releases the error history snapshot. Return ILLEGAL REQUEST/OPERATION IN PROGRESS if another I_T nexus tries to use mode 1Ch while an error history snapshot exists - this prohibits retrieving the snapshot from a different I_T nexus, but ensures a different I_T nexus cannot fetch part of a snapshot and have it released in the middle by the first I_T nexus. Moved all the rules of what the device server must do into the command descriptions, rather than implying them by specifying what the application client must do. Clarified usage of OFFSET BOUNDARY and BUFFER CAPACITY fields for all modes, especially the new one (with 3 options for CAP WG to choose between). Defined that the CLR bit in WRITE BUFFER only means the application client error history, if any, is ignored, not that all fields (like length fields) are ignored.

Related documents

spc4r05a - SCSI Primary Commands - 4 (SPC-4) revision 5a

Overview

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.

Suggested changes

4.5.6 Sense key and sense code definitions

Table 28 - ASC and ASCQ assignments

New:

ASC=2Ah, ASCQ=0Ah

Command sets=all that support the READ BUFFER command, which is all but B (RBC)

Description=ERROR HISTORY SNAPSHOT RELEASED

Editor’s Note 1: Also add to annex C

5.11 Error history

5.11.1 Error history overview [new section, changes not highlighted]

Error history is optional data collected by a logical unit to aid in troubleshooting errors.

The READ BUFFER command (see 6.14.9) provides a method for retrieving error history from the logical unit.

The WRITE BUFFER command (see 6.36.14) provides a method for inserting application client error history into the error history and for clearing the error history.

The format of the application client error history is defined by the vendor of the application client. The format of the error history, including the format of any application client error history included in the error history, is defined by the vendor of the logical unit.

[end of all-new section]

5.11.2 Retrieving error history with the READ BUFFER command [new section, changes not highlighted]

Device servers may allow the error history to be retrieved using a sequence of READ BUFFER commands on one I_T nexus.

To retrieve the complete error history, an application client:

1) Creates an error history snapshot, if one does not already exist, and retrieves the error history directory by sending a READ BUFFER command with:
   A) the MODE field set to 1Ch (i.e., error history);
   B) the BUFFER ID field set to 00h (i.e., error history directory);
   C) the BUFFER OFFSET field set to 000000h; and
   D) the ALLOCATION LENGTH field set to at least 2 056 (i.e., large enough to transfer the complete error history directory);
2) Retrieves the error history. The application client uses a data-in buffer size that is a multiple of the offset boundary indicated in the READ BUFFER descriptor (see 6.16.5). For each buffer ID indicated in the error history directory in the range of 01h through FEh, the application client sends one or more READ BUFFER commands with:
   A) the MODE field set to 1Ch (i.e., error history); and
   B) the BUFFER ID field set to the buffer ID (i.e., error history data); and
   C) the ALLOCATION LENGTH field set to the size of the data-in buffer;

   For the first READ BUFFER command for a particular buffer ID, the application client sets the BUFFER OFFSET field to 000000h. If the number of bytes returned does not equal the allocation length and the total number of bytes returned from the buffer ID does not equal the maximum available length indicated in the error history directory, there is more data in the buffer ID and the application client sends another READ BUFFER command with:

   A) the MODE field set to 1Ch (i.e., error history); and
   B) the BUFFER ID field set to the buffer ID (i.e., error history data).
   C) the BUFFER OFFSET field set to the previous buffer offset plus the previous allocation length; and
   D) the ALLOCATION LENGTH field set to the size of the data-in buffer.
3) Releases the error history snapshot by sending a READ BUFFER command with:
   A) the MODE field set to 1Ch (i.e., error history);
   B) the BUFFER ID field set to FFh (i.e., release error history snapshot);
   C) the BUFFER OFFSET field set to 000000h; and
   D) the ALLOCATION LENGTH field set to 000000h.

While an error history snapshot exists, the device server may or may not record error history for events that it detects but shall record application client error history received in the WRITE BUFFER command. After an error history snapshot is released, the device server shall resume recording error history for events that it detects.

The device server shall release the error history snapshot when:

a) the application client issues a READ BUFFER command with
   A) the MODE field set to 1Ch (i.e., retrieve error history);
   B) the BUFFER ID field set to FFh (i.e., release error history snapshot);
   C) the BUFFER OFFSET field set to 000000h; and
   D) the ALLOCATION LENGTH set to 000000h (see 6.16.9.4);
b) a vendor specific timer expires;
c) a power on occurs;
d) a hard reset occurs;
e) a logical unit reset occurs; or
f) an I_T nexus loss occurs of the I_T nexus that created the error history snapshot.

If the vendor-specific timer expires, the device server shall establish a unit attention condition for the I_T nexus for which the error history snapshot was created with the additional sense code set to ERROR HISTORY SNAPSHOT RELEASED.

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

[end of all-new section]

5.11.3 ~~5.11~~ Adding aApplication client ~~logging~~error history with the WRITE BUFFER command

~~Application client logging is a method the~~An application client may use the WRITE BUFFER command to add ~~use to store~~ application client detected error history ~~information in a logical unit's non-volatile storage (see 6.38.14). The information the application client sends to the logical unit is appended to an application error log~~to the error history collected by a logical unit. The application client error history ~~information~~ may be recovered as part of the error history (see 5.11.2) or by means outside the scope of this standard and is not used for any logical unit related error recovery.

~~A log~~ Error history that contains a mix of application client error history ~~information~~ and logical unit error history ~~information~~ may be used to correlate an application client-detected error with ~~any~~ errors detected internally by~~to~~ the logical unit.~~This does not replace the vendor specific methods for collecting and analyzing engineering data, but provides a vendor independent way of correlating error logs.~~

Application clients should minimize the amount of error history ~~information that is requested to be logged~~ they store to prevent ~~log~~error history overflows (see 6.38.14).

[resultant wording without change bars]

An application client may use the WRITE BUFFER command to add application client detected error history to the error history collected by a logical unit. The application client error history may be recovered as part of the error history (see 5.11.2) or by means outside the scope of this standard and is not used for any logical unit related error recovery.

Error history that contains a mix of application client error history and logical unit error history may be used to correlate an application client-detected error with errors detected internally by the logical unit.

Application clients should minimize the amount of error history they store to prevent error history overflows (see 6.38.14).

5.11.4 Clearing error history with the WRITE BUFFER command [new section, changes not highlighted]

An application client clears the portions of the error history that the device server allows to be cleared by issuing a WRITE BUFFER command (see 6.36.14) with:

a) the BUFFER ID field set to 1Ch (i.e., download error history);
b) the BUFFER OFFSET field set to 000000h;
c) the PARAMETER LIST LENGTH field set to 00001Ah;
d) in the parameter list, the CLR bit set to one; and
e) in the parameter list, the other fields each set to zero.

Clearing error history shall not affect the contents of the error history snapshot, if any, that has been created with the READ BUFFER command (see 5.11.2).

[end of all-new section]

6.16 READ BUFFER command

6.16.1 READ BUFFER command introduction

The READ BUFFER command (see table 135) is used in conjunction with the WRITE BUFFER command for:

a) Testing logical unit buffer memory;
b) Testing the integrity of the service delivery subsystem; ~~and~~
c) Downloading microcode (see 5.15)~~.~~; and
d) Retrieving error history.

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 135.

Table 135 — READ BUFFER MODE field

Code	Description	Reference
00h	Combined header and data [a]	6.16.2
01h	Vendor specific [a]	6.16.3
02h	Data	6.16.4
03h	Descriptor	6.16.5
04h - 09h	Reserved	
0Ah	Echo buffer	6.16.6
0Bh	Echo buffer descriptor	6.16.7
0Bh	Reserved	
1Ah	Enable expander communications protocol and Echo buffer	6.16.8
1Bh	Reserved	
1Ch	Error history	5.11 and 6.16.9
1Dh - 1Fh	Reserved	

[a] Modes 00h and 01h are not recommended.

If the mode is not set to one, the ALLOCATION LENGTH field is defined in 4.3.4.6.

6.16.5 Descriptor mode (03h)

In this mode, a maximum of four bytes of READ BUFFER descriptor information is returned. ~~The device server shall return the descriptor information for the buffer specified by the BUFFER ID field (see the description of the buffer ID in 6.16.4). If there is no buffer associated with the specified buffer ID, the device server shall return all zeros in the READ BUFFER descriptor.~~ The BUFFER OFFSET field is reserved in this mode. The allocation length should be set to four or greater. The READ BUFFER descriptor is defined as shown in table 138.

Editor's Note 2: Something may replace the stricken text to explain how the BUFFER ID field is used based on the option selected by the CAP WG below.

Table 136 — READ BUFFER descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	OFFSET BOUNDARY							
1	(MSB)	BUFFER CAPACITY						
3								(LSB)

~~The OFFSET BOUNDARY field returns the boundary alignment within the selected buffer for subsequent WRITE BUFFER and READ BUFFER commands. The value contained in the OFFSET BOUNDARY field shall be interpreted as a power of two.~~

~~The value contained in the BUFFER OFFSET field of subsequent WRITE BUFFER and READ BUFFER commands should be a multiple of $2^{\text{offset boundary}}$ as shown in table 139.~~

~~Table 139 defines the OFFSET BOUNDARY field.~~

~~**Table 139 — Buffer offset boundary field**~~

~~Offset boundary~~	~~$2^{\text{offset boundary}}$~~	~~Buffer offsets~~
~~0h~~	~~$2^0 = 1$~~	~~Byte boundaries~~
~~1h~~	~~$2^1 = 2$~~	~~Even-byte boundaries~~
~~2h~~	~~$2^2 = 4$~~	~~Four-byte boundaries~~
~~3h~~	~~$2^3 = 8$~~	~~Eight-byte boundaries~~
~~4h~~	~~$2^4 = 16$~~	~~16-byte boundaries~~
~~.~~	~~.~~	~~.~~
~~FFh~~	~~Not applicable~~	~~0 is the only supported buffer offset~~

The OFFSET BOUNDARY field applies to READ BUFFER commands using the following modes:

a) 02h (i.e., Data); and
b) 1Ch (i.e., Error history),

and WRITE BUFFER commands using the following modes:

a) 02h (i.e., Data);
b) 06h (i.e., Download microcode with offsets and activate);
c) 07h (i.e., Download microcode with offsets, save, and activate);
d) 0Eh (i.e., Download microcode with offsets, save, and defer active); and
e) 1Ch (i.e., Download application client error history).

Editor's Note 3: The phrases "descriptor information for the buffer ID specified by the BUFFER ID field" and "boundary alignment within the specified buffer" are unclear. The rest of the READ BUFFER and WRITE BUFFER rules assume the buffer ID definition depends on the scope of the selected mode. This text assumes the mode value is only considered after the buffer ID is selected, and a certain buffer ID is the same size and has the same alignment restrictions in all modes. That is not how new READ BUFFER mode 1Ch wants to use the buffer ID field.

[Choose Option A, B, or C below]

[Option A: Simplest approach - the Buffer ID field is totally irrelevant. The offset boundary applies to all the aforementioned modes regardless of the buffer ID used.]

The boundary alignment applies regardless of the buffer specified by the BUFFER ID field.

[Option B: The Buffer ID field does matter and applies to modes 02h, 06h, 07h, and 0Eh. For the new mode 1Ch, it is not used.]

For mode 1Ch, the boundary alignment applies regardless of the buffer specified by the BUFFER ID field. For modes other than 1Ch to which the OFFSET BOUNDARY field applies, the boundary alignment applies only to the buffer specified by the BUFFER ID field.

[Option C: The Buffer ID field only applies to mode 02h. For all other modes, it is not used.]

For mode 02h (i.e., Data), the boundary alignment applies only to the buffer specified by the BUFFER ID field. For modes other than 02h to which the OFFSET BOUNDARY field applies, the boundary alignment applies regardless of the buffer specified by the BUFFER ID field.

Table 139 defines the OFFSET BOUNDARY field.

Table 139 — OFFSET BOUNDARY field

Code	Supported BUFFER OFFSET field values
n = 00h to FEh	Multiples of 2^n bytes (e.g., a) 00h means multiples of 2^0 (i.e., 1) bytes (i.e., no restrictions); b) 01h means multiples of 2^1 (i.e., 2) bytes (i.e., even offsets); and c) 02h means multiples of 2^2 (i.e., 4) bytes)
FFh	000000h is the only supported buffer offset

~~The BUFFER CAPACITY field shall return the size of the selected buffer in bytes.~~

The BUFFER CAPACITY field applies to READ BUFFER commands using the following modes:

a) 02h (i.e., Data),

[Choose Option Y or Z below]

[Option Y: Buffer Capacity applies only to mode 02h]

and WRITE BUFFER commands using the following modes:

a) 02h (i.e., Data).

[Option Z: Buffer Capacity applies to download microcode modes]

and WRITE BUFFER commands using the following modes:

a) 02h (i.e., Data);
b) 06h (i.e., Download microcode with offsets and activate);
c) 07h (i.e., Download microcode with offsets, save, and activate); and
d) 0Eh (i.e., Download microcode with offsets, save, and defer active).

[Choose Option A, B, or C below. If option Y is chosen, options B and C are equivalent.]

[Option A: Simplest approach - the Buffer ID field is totally irrelevant. The offset applies to all the aforementioned modes regardless of the buffer ID used.]

The buffer capacity applies regardless of the buffer specified by the BUFFER ID field.

The BUFFER CAPACITY field indicates the maximum size in bytes of the buffer.

[Option B: The Buffer ID field does matter for modes 02h, 06h, 07h, and 0Eh. For 1Ch, it is not used.]

For mode 1Ch, the buffer capacity applies regardless of the buffer specified by the BUFFER ID field. For modes other than 1Ch, the buffer capacity applies only to the buffer specified by the BUFFER ID field.

The BUFFER CAPACITY field indicates the maximum size in bytes of the buffer.

[Option C: The Buffer ID field only applies to mode 02h. For all other modes, it is not used.]

For mode 02h (i.e., Data), the buffer capacity applies only to the buffer specified by the BUFFER ID field. For modes other than 02h, the buffer capacity does not apply.

The BUFFER CAPACITY field indicates the maximum size in bytes of the buffer.

NOTE 25 - In a system employing multiple application clients, a buffer may be altered between the WRITE BUFFER and READ BUFFER commands by another application client. Buffer testing applications should ensure that only a single application client is active. Use of reservations to all logical units on the device may be helpful in avoiding buffer alteration between these two commands.

Editor's Note 4: This note seems totally misplaced. It applies to modes like the data mode and echo buffer modes, but not the download microcode or error history modes. This proposal does not suggest where to move it, though.

6.16.9 Error history mode (1Ch) [all new]

6.16.9.1 Error history overview

This mode is used to retrieve error history (see 5.11).

If the device server receives a READ BUFFER command with the MODE field set to 1Ch from a different I_T nexus while an error history snapshot exists, it shall terminate the command with CHECK CONDITION status with the sense key set to ILLEGAL REQUEST and the additional sense code set to OPERATION IN PROGRESS.

The BUFFER ID field (see table 197) specifies the action and the parameter data, if any, that the device server shall return.

Table 197 — BUFFER ID field

Code	Description	BUFFER OFFSET field	Reference
00h	Error history directory	Shall be set to 00h	6.16.9.2
01h - FEh	Error history data	Honored	6.14.9.3
FFh	Release error history snapshot	Ignored	6.16.9.4

The BUFFER OFFSET field specifies the byte offset from the start of the buffer specified by the BUFFER ID field from which the device server shall return data. 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 device server shall terminate the READ BUFFER command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

6.16.9.2 Error history directory

When the BUFFER ID field is set to 00h, the device server shall:

a) if an error history snapshot does not already exist for the I_T nexus, take an error history snapshot of the error history data; and
b) return an error history directory (see table 198) indicating the error history data that is available for retrieval.

If an error history snapshot already exist for the I_T nexus, the device server shall not consider this an error.

The error history directory is defined in table 198.

Table 198 — Error history directory

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB)	T10 VENDOR IDENTIFICATION						
7								(LSB)
8	VERSION							
9	Reserved							CLR_SUP
10	Reserved							
13								
14	DIRECTORY LENGTH (n - 15)							
15								
Error history directory list								
16	Error history directory entry (first)(see table 199)							
23								
...	...							
n - 7	Error history directory entry (last)(see table 199)							
n								

The T10 VENDOR IDENTIFICATION field contains eight bytes of left-aligned ASCII data (see 4.4.1) identifying the manufacturer of the logical unit. 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).

NOTE 26 - The T10 VENDOR IDENTIFICATION field may contain a different value than the VENDOR IDENTIFICATION field in the standard INQUIRY data (see 6.4.2) (e.g., this field may indicate a disk drive component vendor while the standard INQUIRY data indicates the original equipment manufacturer).

The VERSION field indicates the version and format of the vendor-specific error history. The VERSION field is assigned by the vendor indicated in the T10 VENDOR IDENTIFICATION field.

A clear support (CLR_SUP) bit set to one indicates that the CLR bit is supported in the WRITE BUFFER command download error history mode (see 6.36.14). A CLR_SUP bit set to zero indicates that the CLR bit is not supported.

The DIRECTORY LENGTH field indicates the number of error history directory list bytes available to be transferred. This value shall not be altered even if the allocation length is not sufficient to transfer the entire error history directory list.

The error history directory list contains an error history directory entry (see table 199) for each supported buffer ID in the range of 00h through FEh. The first entry shall be for buffer ID 00h 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 FFh.

The error history directory entry is defined in table 199.

Table 199 — Error history directory entry

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

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

The MAXIMUM AVAILABLE LENGTH field indicates the maximum number of data bytes contained in the buffer indicated by the SUPPORTED BUFFER ID field. The actual number of bytes available for transfer may be smaller.

6.16.9.3 Error history data

When the BUFFER ID field is set to a value in the range of 01h to FEh that is indicated as supported in the error history directory (see 6.16.9.2) and the BUFFER OFFSET field is set to a valid value, the device server shall return error history data (see table 200) from the error history snapshot from the specified buffer at the specified buffer offset.

The amount of error history data in the specified buffer shall be less than or equal to the number of bytes indicated by the MAXIMUM AVAILABLE LENGTH field in the error history directory (see 6.16.9.2).

If:

a) an error history snapshot does not exist for the I_T nexus;
b) the BUFFER ID field is set to a value in the range of 01h to FEh that is not indicated in the error history directory; or
c) the BUFFER OFFSET field is not set to retrieve the next contiguous data within the specified buffer ID,

then the device server shall shall terminate the READ BUFFER command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to INVALID FIELD IN CDB.

Editor’s Note 5: 2/28 go ahead and allow reading of the buffer in any order once selected. Let the directory be re-fetched without taking a new snapshot - only takes a new snapshot if one is not in place.

If:

a) an error history snapshot exists for the I_T nexus;
b) the BUFFER ID field is set to a value in the range of 01h to FEh that is indicated in the error history directory;
c) there was a previous READ BUFFER command retrieving error history data for this error history snapshot, and the BUFFER ID field is set to a different value than in that command; and
d) the previous READ BUFFER command did not obtain the last data from the previous buffer,

then the device server shall shall terminate the READ BUFFER command with CHECK CONDITION status, with the sense key set to ILLEGAL REQUEST, and the additional sense code set to COMMAND SEQUENCE ERROR.

Editor’s Note 6: Perhaps the contiguous rule generating INVALID FIELD IN CDB and the ”fetch all data from a buffer” COMMAND SEQUENCE ERROR are not necessary/helpful any more. The device server shouldn't try to protect the application from shooting itself in the foot and not retrieving all the data.

The error history data is defined in table 200.

Table 200 — Error history data

Byte\Bit	7	6	5	4	3	2	1	0
0	Vendor specific							
n								

The contents of the error history data are vendor specific.

6.16.9.4 Release error history snapshot

When the BUFFER ID field is set to FFh, the device server shall:

a) release the error history snapshot, if any; and
b) not transfer any data.

If an error history snapshot does not exist for the I_T nexus, the device server shall not consider this an error.

[End of all-new section]

6.38 WRITE BUFFER command

6.38.1 WRITE BUFFER command introduction

The WRITE BUFFER command (see table 210) is used in conjunction with the READ BUFFER command as a function for:

a) Testing logical unit buffer memory;
b) Testing the integrity of the service delivery subsystem;
c) Downloading microcode (see 5.15); and
d) Downloading application ~~logs~~client error history (see 5.11).

Table 210 — WRITE BUFFER command

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

This 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 211.

Table 211 — WRITE BUFFER MODE field

Code	Description	Reference
00h	Combined header and data [a]	6.36.2
01h	Vendor specific [a]	
02h	Data	6.36.3
03h	Reserved	
04h	Download microcode and activate	5.15 and 6.36.4
05h	Download microcode, save, and activate	5.15 and 6.36.5
06h	Download microcode with offsets and activate	5.15 and 6.36.6
07h	Download microcode with offsets, save and activate	5.15 and 6.36.7
08h - 09h	Reserved	
0Ah	Echo buffer	6.36.8
0Bh - 0Dh	Reserved	
0Eh	Download microcode with offsets,save, and defer activate	5.15 and 6.36.9
0Fh	Activate deferred microcode	5.15 and 6.36.10
1Ah	Enable expander communications protocol and Echo buffer	6.36.11
1Bh	Disable expander communications protocol	6.36.12
1Ch	Download application ~~log~~ client error history	5.15 and 6.36.13
1Dh - 1Fh	Reserved	

[a] Modes 00h and 01h are not recommended.

6.38.14 Download application ~~log~~client error history mode (1Ch)

In this mode the device server transfers data from the application client and stores it in ~~an application log~~the error history (see 5.11). The format of the application ~~log data~~client error history is as specified in table 212.

The BUFFER ID field and BUFFER OFFSET field shall be ignored in this mode.

Upon successful completion of a WRITE BUFFER command the data shall be appended to the ~~application log~~error history in a format determined by the logical unit.

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 ~~application log~~error history. If the PARAMETER LIST LENGTH field specifies a transfer that exceeds the ~~application log's~~error history 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 212 — Application ~~log data WRITE BUFFER~~client error history format

Byte\Bit	7	6	5	4	3	2	1	0
0	T10 VENDOR IDENTIFICATION							
7								
8	(MSB)	ERROR TYPE						
9								(LSB)
10	~~Reserved~~							
10	Reserved							CLR
11	Reserved							
12	(MSB)	TIME~~-~~STAMP						
17								(LSB)
18	Reserved							
19								
20	Reserved				CODE SET			
21	ERROR LOCATION FORMAT							
22	(MSB)	ERROR LOCATION LENGTH (m - 25)						
23								(LSB)
24	(MSB)	~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY LENGTH (n - m)						
25								(LSB)
26	(MSB)	ERROR LOCATION						
m								(LSB)
m+1	~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY							
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 D and on the T10 web site (http://www.T10.org).

The ERROR TYPE field (see table 213) specifies the error detected by the application client.

Table 213 — 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 (see SAM-4).
0005h - 7FFFh	Reserved
8000h - FFFFh	Vendor specific

A CLR bit set to one specifies that the device server shall clear the portions of the error history that the device server allows to be cleared, and that any application client error history also specified in the parameter list shall be ignored. A CLR bit set to zero specifies that the device server shall not clear the error history.

The TIME-STAMP field ~~shall~~should contain:

a) The number of milliseconds that have elapsed since midnight, 1 January 1970 UT (see 3.1.130); or
b) Zero, if the application client is not able to determine the UT of the log entry.

The CODE SET field (see table 214) specifies the code set used for the application ~~log information~~client error history and shall only apply to information contained in the ~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY field.

NOTE 34 - The CODE SET field is intended to be an aid to software that displays the application ~~log information~~client error history.

Table 214 — CODE SET field

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

The ERROR LOCATION FORMAT field (see table 215) specifies the format of the ERROR LOCATION field.

Table 215 — ERROR LOCATION FORMAT field

Code	Description
00h	No error history location specified by the application client
01h	For direct-access block devices (see SBC-3 and RBC), the ~~The~~ ERROR LOCATION field specifies the logical block address (e.g., LBA) associated with ~~the error information contained within the application log~~specified application client error history. For other peripheral device types, this code is reserved.
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~~ ERROR LOCATION LENGTH field set to zero specifies that there is no error location information.

The ~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY LENGTH field specifies the length of the ~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY field. The ~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY LENGTH field value shall be a multiple of four. A ~~vendor specific length of~~ APPLICATION CLIENT ERROR HISTORY LENGTH set to zero specifies there is no ~~vendor specific information~~application client error history.

The ERROR LOCATION field specifies the location at which the application client detected the error, in the format specified by the ERROR LOCATION FORMAT field.

The ~~VENDOR SPECIFIC~~APPLICATION CLIENT ERROR HISTORY field ~~provides~~contains ~~vendor specific information on the error~~vendor specific application client error history (see 5.11.1).