To: T10 Technical Committee
From: Gerry Houlder, Seagate Technology (gerry.houlder@seagate.com) and
Rob Elliott, HP (elliott@hp.com)
Date: 4 April 2007
Subject: 06-362r5 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.

Revision 5 (4 April 2007) Incorporated changes requested at March 2007 CAP WG meeting and 3/27/2007 conference call on the proposal. Separated concept of the “error history I_T nexus” from the snapshot and added a “clear error history I_T nexus” mode FEh so multiple I_T nexuses can retrieve the same snapshot (one at a time). Reserved buffer IDs F0h - FDh for future enhancements. Allow device server to take error history snapshots itself; report the source of the snapshot in the error history directory. Increased error history directory header size from 16 to 32 bytes to allow for future enhancements. Added buffer ID 02h and 03h to preempt the error history I_T nexus. Reserved buffer IDs 04h-0Fh for future enhancements. Deleted note recommending using reservations along with WRITE BUFFER/READ BUFFER commands and added note in READ BUFFER modes 00h and 02h warning that the buffers are shared across application clients and I_T nexuses. Allow the vendor-specific timer to either clear the error history I_T nexus only, or clear it and release the snapshot; report unique additional sense codes for each of those.

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

Add new additional sence codes:

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

ASC=2Ah, ASCQ=0Ah Description=ERROR HISTORY I_T NEXUS CLEARED

ASC=2Ah, ASCQ=0Bh 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 (see 5.11.2).

The WRITE BUFFER command (see 6.36.14) provides a method for inserting application client error history into the error history (see 5.11.3) and for clearing the error history (see 5.11.4).

The format of the application client error history is defined by the manufacturer 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 manufacturer 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.

Error history is returned using error history snapshots. An error history snapshot is the contents of the error history at a specific point in time, created by the device server at vendor-specific times or by the application client using the READ BUFFER command with certain buffer IDs.

The I_T nexus being used to retrieve error history snapshot is called the error history I_T nexus. Only one I_T nexus is allowed to retrieve error history snapshot at a time.

To retrieve the complete error history, an application client uses one I_T nexus to:

1) Create an error history snapshot if one does not already exist, establish the I_T nexus as the error history I_T nexus, and retrieve the error history directory by sending a READ BUFFER command with (see 6.16.9.2):
   A) the MODE field set to 1Ch (i.e., error history);
   B) the BUFFER ID field set to 00h (i.e., error history directory), 01h (i.e., error history directory with new snapshot), 02h (i.e., error history directory with new error history I_T nexus), or 03h (i.e., error history directory with new error history I_T nexus and snapshot);
   C) the BUFFER OFFSET field set to 000000h; and
   D) the ALLOCATION LENGTH field set to at least 2 088 (i.e., large enough to transfer the complete error history directory);

   The application client should only set the BUFFER ID field to 02h or 03h if it has knowledge obtained by means outside the scope of this standard that the error history I_T nexus is no longer valid.

2) Retrieve 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 10h through EFh, the application client sends one or more READ BUFFER commands with (see 6.16.9.3):

A) the MODE field set to 1Ch (i.e., error history); and
B) the BUFFER ID field set to the buffer ID (i.e., an error history data buffer); 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., an error history data buffer).
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) Clear the error history I_T nexus and, depending on the buffer ID, release the error history snapshot by sending a READ BUFFER command with (see 6.16.9.4 and 6.16.9.5):
   A) the MODE field set to 1Ch (i.e., error history);
   B) the BUFFER ID field set to FEh (i.e., clear error history I_T nexus) or FFh (i.e., clear error history I_T nexus and release 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:

a) shall not modify the error history snapshot to reflect any changes to the error history;
b) may or may not record events that it detects into the error history; and
c) shall record application client error history received in the WRITE BUFFER command into the error history.

The device server shall clear the error history I_T nexus and not release the error history snapshot when:

a) the application client issues a READ BUFFER command using the error history I_T nexus with (see 6.16.9.4):
   A) the MODE field set to 1Ch (i.e., retrieve error history);
   B) the BUFFER ID field set to FEh (i.e., clear error history I_T nexus);
   C) the BUFFER OFFSET field set to 000000h; and
   D) the ALLOCATION LENGTH set to 000000h;

   or

b) an I_T nexus loss occurs of the error history I_T nexus.

The device server shall clear the error history I_T nexus and release the error history snapshot when (see 6.16.9.5):

a) the application client issues a READ BUFFER command using the same I_T nexus that was used to establish the snapshot 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;
b) a power on occurs;
c) a hard reset occurs; or
d) a logical unit reset occurs.

The device server shall implement a vendor specific timer for error history snapshot retrieval. If the vendor specific timer expires, the device server shall either:

a) clear the error history I_T nexus; and
b) establish a unit attention condition for the error history I_T nexus with the additional sense code set to ERROR HISTORY I_T NEXUS CLEARED;

or:

a) clear the error history I_T nexus;
b) release the error history snapshot; and
c) establish a unit attention condition for the error history I_T nexus with the additional sense code set to ERROR HISTORY SNAPSHOT RELEASED.

The device server may replace or release an error history snapshot that it takes at any time that the error history I_T nexus is clear. The device server shall not replace or release the error history snapshot while the error history I_T nexus is not clear.

After an error history snapshot is released, the device server shall resume recording error history for events that it detects.

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 a~~A~~pplication client ~~logging~~error history with the WRITE BUFFER command

~~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 6.38.14). The information the application client sends to the logical unit is appended to an application error log. The application client error information may be recovered by means outside the scope of this standard and is not used for any logical unit related error recovery.~~

~~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 internall~~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 information that is requested to be logged to prevent log overflows.~~

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.

6.16.2 Combined header and data mode (00h)

In this mode, a four-byte header followed by data bytes is returned to the application client in the Data-In Buffer.

The allocation length should be set to four or greater. The BUFFER ID and the BUFFER OFFSET fields are reserved.

The four-byte READ BUFFER header (see table 137) is followed by data bytes from the buffer.

Table 136 — READ BUFFER descriptor

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved							
1	(MSB)			BUFFER CAPACITY				
3								(LSB)
4				Data				
n								

The BUFFER CAPACITY field specifies the total number of data bytes available in the buffer. The buffer capacity is not reduced to reflect the actual number of bytes written using the WRITE BUFFER command Combined header and data mode (see 6.38.2). The relationship between the BUFFER CAPACITY field and the CDB ALLOCATION LENGTH field is defined in 4.3.4.6.

Following the READ BUFFER header, the device server shall transfer data from the buffer.

NOTE 1 - The buffer is shared by all application clients and I_T nexuses. If one application client writes the buffer, a second application client writes the buffer, and then the first application client reads the buffer, the read retrieves the data from the second application client.

6.16.4 Data mode (02h)

In this mode, the Data-In Buffer is filled only with logical unit buffer data. The BUFFER ID field specifies a buffer within the logical unit from which data shall be transferred. The vendor assigns buffer ID codes to buffers within the logical unit. Buffer ID zero shall be supported. If more than one buffer is supported, then additional buffer ID codes shall be assigned contiguously, beginning with one. Buffer ID code assignments for the READ BUFFER command Data mode shall be the same as for the WRITE BUFFER command Data mode (see 6.38.3). If an unsupported buffer ID code is selected, 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.

NOTE 2 - Each buffer is shared by all application clients and I_T nexuses. If one application client writes the buffer, a second application client writes the buffer, and then the first application client reads the buffer, the read retrieves the data from the second application client.

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 6.16.5). 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.

6.16.5 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.

Table 137 — 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~~

Table 139 defines the OFFSET BOUNDARY field.

Table 139 — OFFSET BOUNDARY field

Code	Supported BUFFER OFFSET field values for READ BUFFER and WRITE BUFFER commands with modes honoring this field
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

Editor's Note 2: The following READ BUFFER modes honor the offset boundary and buffer capacity: 02h (Data)

Editor's Note 3: The following READ BUFFER modes don't use the BUFFER OFFSET field: 00h (combined header and data), 01h (vendor specific), 03h (descriptor), 0Ah (echo buffer), 0Bh (echo buffer descriptor), 1Ah (echo buffer)

Editor's Note 4: The following WRITE BUFFER modes honor the offset boundary and buffer capacity: 02h (Data), 06h (download with offsets and activate), 07h (download with offsets, save and activate), 0Eh (download with offsets, save, and defer activate).

Editor's Note 5: The following WRITE BUFFER modes don't use the BUFFER OFFSET field: 00h (combined header and data), 01h (vendor specific), 04h (download and activate), 05h (download, save, and activate), 0Ah (echo buffer), 0Fh (activate), 1Ah (ECP & echo buffer), 1B (disable ECP), 1Ch (application log). 1Ch has its own capacity rules.

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

a) 02h (i.e., Data)(see 6.16.4); and
b) 1Ch (i.e., Error history)(see 6.16.9).

and WRITE BUFFER commands using the following modes:

a) 02h (i.e., Data)(see 6.36.3);
b) 06h (i.e., Download microcode with offsets and activate)(see 6.36.6);
c) 07h (i.e., Download microcode with offsets, save, and activate)(see 6.36.7); and
d) 0Eh (i.e., Download microcode with offsets, save, and defer active)(see 6.36.9).

For mode 02h (i.e., Data), the boundary alignment indicated by the OFFSET BOUNDARY field 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.

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

The BUFFER CAPACITY field indicates the maximum size in bytes of the buffer specified by the BUFFER ID field for.

a) the READ BUFFER command with mode 02h (i.e., Data); and
b) the WRITE BUFFER command with mode 02h (i.e., Data).

~~NOTE 25 - In a system employing multiple application clients, a buffer in modes 00h (i.e., Combined header and data), 02h (i.e., Data), and 0Ah (echo 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 6: This note does not belong here, and its advice to use [persistent] reservations to run WRITE BUFFER/READ BUFFER commands is unlikely to be followed. It applies to modes 00h, 02h, and 0Ah. Mode 0Ah reports ECHO BUFFER OVERWRITTEN if a different I_T nexus than the one that wrote it tries to read the data, while the other two just return the most recently written data. So, a simplified version of the note is added to READ BUFFER modes 00h and 02h.

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 is unable to process a READ BUFFER command with the MODE field set to 1Ch because of a vendor specific condition, it 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.

If:

a) an error history snapshot exists;
b) an error history I_T nexus exists (i.e., the device server has reserved access to the error history snapshot for a particular I_T nexus);
c) and the device server receives a READ BUFFER command with the MODE field set to 1Ch from a different I_T nexus,

then 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 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	Error history directory with new snapshot	Shall be set to 00h	6.16.9.2
02h	Error history directory with new error history I_T nexus	Shall be set to 00h	6.16.9.2
03h	Error history directory with new error history I_T nexus and snapshot	Shall be set to 00h	6.16.9.2
04h - 0Fh	Reserved		
10h - EFh	Error history data buffer	Honored	6.16.9.3
F0h - FDh	Reserved		
FEh	Clear error history I_T nexus	Ignored	6.16.9.4
FFh	Clear error history I_T nexus and release snapshot	Ignored	6.16.9.5

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 indicated 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

If:

a) there is no error history I_T nexus;
b) the BUFFER ID field is set to 00h or 01h and the I_T nexus performing the READ BUFFER command is the error history I_T nexus;
c) the BUFFER ID field is set to 02h or 03h,

then the device server shall:

a) if an error history snapshot does not already exist, take an error history snapshot of the error history;
a) if an error history snapshot already exists and the BUFFER ID field is set to 00h or 02h, preserve the error history snapshot;
a) if an error history snapshot already exists and the BUFFER ID field is set to 01h or 03h, discard the error history snapshot and take another error history snapshot of the error history;
b) set the error history I_T nexus to the I_T nexus being used for the READ BUFFER command; and
c) return an error history directory (see table 198) indicating the error history that is available for retrieval and is stored in the error history snapshot.

If an error history snapshot already exists, 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	T10 VENDOR IDENTIFICATION							
7								
8	VERSION							
9	Reserved					EHS_SOURCE		CLR_SUP
10	Reserved							
29								
30	DIRECTORY LENGTH (n - 15)							
31								
Error history directory list								
32	Error history directory entry (first)(see table 200)							
39								
...	...							
n - 7	Error history directory entry (last)(see table 200)							
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.

Table 199 defines the EHS_SOURCE field.

Table 199 — EHS_SOURCE field

Code	Description
00b	Error history snapshot was created by the device server and was not created due to processing a READ BUFFER command
01b	Error history snapshot was created due to processing of this READ BUFFER command
10b	Error history snapshot was created due to processing of a previous READ BUFFER command
11b	Reserved

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 200) for each supported buffer ID in the range of 00h through EFh. 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 IDs greater than or equal to F0h.

The error history directory entry is defined in table 200.

Table 200 — 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 buffer

If:

a) the I_T nexus performing the READ BUFFER command is the error history I_T nexus;
b) the BUFFER ID field is set to a value in the range of 10h to EFh that is indicated as supported in the error history directory (see 6.16.9.2); and
c) the BUFFER OFFSET field is set to a valid value,

then the device server shall return error history (see table 201) from the error history snapshot from the specified buffer at the specified buffer offset.

The amount of error history 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).

The error history data buffer is defined in table 201.

Table 201 — Error history data buffer

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

The contents of the error history data buffer are vendor specific.

6.16.9.4 Clear error history I_T nexus

If:

a) the I_T nexus performing the READ BUFFER command is the error history I_T nexus; and
b) the BUFFER ID field is set to FEh,

then the device server shall:

a) clear the error history I_T nexus, if any; and
b) not transfer any data.

If the error history I_T nexus is already clear, the device server shall not consider this an error.

6.16.9.5 Clear error history I_T nexus and release snapshot

If:

a) the I_T nexus performing the READ BUFFER command is the error history I_T nexus; and
b) the BUFFER ID field is set to FFh,

then the device server shall:

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

If the error history I_T nexus is already clear, 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)							
4	BUFFER OFFSET							
5								(LSB)
6	(MSB)							
7	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~~							
	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).