

## T10/04-186 revision 2

Date: September 16, 2004

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: SPC-3: Standard Data Collection Trigger

## 1 Overview

Debugging a problem in a SCSI environment can be difficult when one part of the system (e.g., application client) detects an error when the failure is caused by a different part of the system (e.g., logical unit).

Most applications and logical units have some of error logging, but correlating the time of an application client event with the time of a logical unit log can be difficult.

Correlating the location of data in a log can be difficult. For example, the application client may complain about "scsi disk 0" while the logical unit only knows a about it's LUN.

In logical units recoverable errors typically generate minimal log information, while catastrophic errors often generate detailed engineering data. A problem that may be viewed as recoverable at the logical unit may cause a catastrophic failure at the application client. Also, there is no general way to automatically trigger collection of more detailed log data in a logical unit.

This proposal defines a way to pass error information to a logical unit. The information would not be used for any error recovery procedures, but would be used in to enhance debug. For example, the logical unit would add the information received from the application client to its existing error log. The error log could then be used to correlate an application client error with any errors internal to the logical unit. This doesn't replace the vendor specific methods for collecting and analyzing engineering data, but provides a vendor independent way of correlating error logs and provides a trigger mechanism for collecting additional log data.

## 2 Solution

## 3 Definitions, symbols, abbreviations, and conventions

### 3.1 Definitions

....

**3.1.1 universal time (UT):** The time at longitude zero, colloquially known as Greenwich Mean Time. See <http://aa.usno.navy.mil/faq/docs/UT.html>.

....

### 3.2 Acronyms

....

UT      Universal Time (see 3.1.1)

## 4 Model common to all device types

.....

### 4.1 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. The information the application client sends to the logical

unit is appended to an application error log. The application client error information is 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 internal 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.

## 4.2 WRITE BUFFER command

### 4.2.1 WRITE BUFFER command introduction

The WRITE BUFFER command (see table 1) is used in conjunction with the READ BUFFER command as a diagnostic function for testing logical unit memory in the target SCSI device and the integrity of the service delivery subsystem. Additional modes are provided for:

- a) downloading microcode;
- b) for downloading and saving microcode; and
- c) [downloading application logs](#).

**Table 1 — WRITE BUFFER command**

Bit Byte	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 77.

**Table 2 — WRITE BUFFER MODE field**

MODE	Description	Implementation requirements
00h	Write combined header and data	Optional
01h	Vendor specific	Vendor specific
02h	Write data	Optional
04h	Download microcode	Optional
05h	Download microcode and save	Optional
06h	Download microcode with offsets	Optional
07h	Download microcode with offsets and save	Optional
0Ah	Echo buffer	Optional
1Ah	Enable expander communications protocol and Echo buffer	Optional
1Bh	Disable expander communications protocol	Optional
<a href="#">1Ch</a>	<a href="#">Download application log</a>	<a href="#">Optional</a>
03h	Reserved	
08h - 09h	Reserved	
0Bh - 19h	Reserved	
<a href="#">1Dh</a> - 1Fh	Reserved	

NOTES

- 1 Modes 00h and 001h are not recommended.
- 2 When downloading microcode with buffer offsets, the WRITE BUFFER command mode should be 06h or 07h.

**4.2.2 Combined header and data mode (00h)**

....

**4.2.3 Vendor specific mode (01h)**

....

**4.2.4 Data mode (02h)**

....

**4.2.5 Download microcode mode (04h)**

....

**4.2.6 Download microcode and save mode (05h)**

....

**4.2.7 Download microcode with offsets (06h)**

....

#### 4.2.8 Download microcode with offsets and save mode (07h)

....

#### 4.2.9 Write data to echo buffer (0Ah)

....

#### 4.2.10 Enable expander communications protocol and Echo buffer (1Ah)

....

#### 4.2.11 Disable expander communications protocol (1Bh)

....

#### 4.2.12 Download application log (1Ch) (new section)

In this mode the device server transfers data from the application client and stores it in an application log. The format of the application log data is as specified in table 3. The BUFFER ID and BUFFER OFFSET fields are ignored in this mode.

Upon successful completion of a WRITE BUFFER command the data shall be appended to the application log.

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. If the PARAMETER LIST LENGTH field specifies a transfer that exceeds the application log's capacity, the device server shall return CHECK CONDITION status and shall set the sense key to ILLEGAL REQUEST with an additional sense code of INVALID FIELD IN CDB.

**Table 3 — Application Log data format**

Bit Byte	7	6	5	4	3	2	1	0
0	(MSB)	VENDOR IDENTIFICATION						(LSB)
7								
8	(MSB)	ERROR TYPE						(LSB)
9								
10		Reserved						
11		Reserved						
12	(MSB)	TIME STAMP						(LSB)
17								
18		Reserved						
19		Reserved						
20		Reserved			CODE SET			
21		ERROR LOCATION FORMAT						
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		VENDOR SPECIFIC						
n								

The VENDOR IDENTIFICATION field contains eight bytes of left-aligned ASCII data (see 4x.4x.1x) identifying the vendor of the application. The vendor identification shall be one assigned by INCITS. A list of assigned vendor identifications is in Annex Dx and on the T10 web site ([www.T10.org](http://www.T10.org)).

The ERROR TYPE field (see table 4) specifies the error detected by the application client.

**Table 4 — ERROR TYPE field values**

Value	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

The TIME STAMP field is represented as a 48-bit unsigned fixed-point number in big-endian format. The TIME STAMP field shall contain the number of milliseconds that have elapsed since midnight, 1 January 1970 UT (see 3.1.1). The application client shall set the TIME STAMP field to zero if it is not able to determine the UT of the log entry.

The CODE SET field specifies the code set used for the application log information (see table 5) and shall only apply to information contained in the VENDOR SPECIFIC field.

NOTE 3 - The CODE SET field is intended to be an aid to software that displays the application log information.

**Table 5 — Code set**

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

The ERROR LOCATION FORMAT field specifies the format (see table 6) of the ERROR LOCATION field.

**Table 6 — ERROR LOCATION FORMAT field values**

Value	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 application log.
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.