

# Logical Unit Self-tests

**To:** T10 Technical committee  
**From:** Mark Evans  
Quantum Corporation  
500 McCarthy Boulevard  
Milpitas, CA USA 95035  
**Phone:** 408-894-4019  
**Fax:** 408-952-3620  
**Email:** mark.evans@quantum.com  
**Date:** 23 July 1999

## 1 Introduction

As the complexity of today's systems increases, a method is needed to quickly identify the root cause of system problems at the end-user's site. Self-test routines implemented in a logical unit (e.g., a hard disk drive) would provide a mechanism to quickly determine whether the logical unit is faulty or indicate that some other component is causing the system to fail. These self-test routines could be performed by a logical unit without removing it from its installation. This proposal recommends changes to SPC-2 to provide these features.

What's new for Revision 3: All comments from the SCSI Working Group of 14 July 1999 have been included. The STPI bit has been removed. It was determined that this is not necessary as the progress indication for a self-test should be returned as normal in response to a REQUEST SENSE command. All references to and models for SEND DIAGNOSTICS and MODE SENSE/SELECT in SPC-2 and SAM-2 were checked to make the descriptions of how the background and foreground self-test modes function be consistent. Unfortunately, the concepts of "background" and "foreground" tasks are not well described or not addressed at all in these standards. Included in what is not specified is how a device server should respond to each command type while performing testing in one of these modes. It is recommended that the specifications developed in this proposal for those functions be included where appropriate in SPC-2.

## 2 Reference documents

Elements from the draft standard SCSI Primary Commands – 2 (SPC-2) Revision 10, 19 May 1999 are referenced in this proposal. For more detail, see the referenced clauses. This draft is available at <ftp://ftp.t10.org/t10/drafts/spc2/spc2-r10.pdf>.

## 3 Overview model

### 3.1 The short and extended self-tests

There are two types of self-tests: a short self-test and an extended self-test. The goal of the short self-test is to quickly identify if the logical unit determines that it is faulty. A goal of the extended self-test routine is to simplify factory testing during integration by having logical units perform more comprehensive testing without application client intervention. A second goal of the extended self-test is to provide a more comprehensive test to validate the results of the short self-test if its results are judged by the application client to be inconclusive.

The criteria for the short self-test are that the self-test has one or more segments and completes in two minutes or less. The criteria for the extended self-test are that it has one or more segments and that the completion time is vendor-specific. Any tests performed in the segments are vendor-specific.

**[EDITOR'S NOTE (mse – 23 Jul 99): It has been recommended that the maximum time be decreased to one second. Is there any discussion about this?]**

The following are examples of segments:

- a) An electrical segment wherein the logical unit tests its own electronics. The tests in this segment are vendor specific, but some examples of tests that might be included in this segment are: a buffer RAM test, a read/write circuitry test, and/or a test of the read/write head elements.
- b) A seek/servo segment wherein a device tests its capability to find and servo on data tracks.
- c) A read/verify scan segment wherein a device performs read scanning of some or all of the medium surface.

The tests performed in the segments may be the same for the short and extended self-tests. The time required by a logical unit to complete its extended self-test is reported in the EXTENDED SELF-TEST ROUTINE COMPLETION TIME field in the Control mode page.

### 3.2 Self-test modes

There are two modes for each type of self-test: a foreground mode and a background mode.

When a device server receives a SEND DIAGNOSTICS command specifying a self-test to be performed in the foreground mode, the device server shall return status for that command after the self-test has been completed. When a device server receives a SEND DIAGNOSTICS command specifying a self-test to be performed in the background mode, the device server shall return status for that command as soon as the command descriptor block has been validated.

While performing a self-test in the foreground mode, the device server shall respond to all commands except INQUIRY, REPORT LUNS, and REQUEST SENSE with CHECK CONDITION status with a sense key of NOT READY and an additional sense code of LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS. It is recommended that, before an application client initiates a self-test in the foreground mode, it should reserve the logical unit.

**[EDITOR'S NOTE (mse – 23 Jul 99): SBC specifies that a device server shall return a RESERVATION CONFLICT status if it receives a FORMAT UNIT command when any other initiator has an extent reservation on that device server. For consistency I think that a device server should respond the same way when an application client issues a SEND DIAGNOSTICS command specifying a self-test to be performed in the foreground mode and any other initiator has an extant reservation. Any comments?]**

**[EDITOR'S NOTE (mse – 23 Jul 99): SBC specifies that, while executing a FORMAT UNIT command, a device server shall respond to a REQUEST SENSE command by returning a CHECK CONDITION and an additional sense code of NOT READY/LOGICAL UNIT NOT READY, FORMAT IN PROGRESS with the sense key specific bytes set for progress indication (unless an error has occurred). For consistency I think that a device server respond in a similar fashion when an application client issues a REQUEST SENSE command while the device server is performing a self-test in the foreground mode (only instead respond with a CHECK CONDITION and an additional sense code of NOT READY/LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS with the sense key specific bytes set for progress indication). Any comments?]**

While performing a self-test in the background mode, the device server shall suspend the self-test to service any other commands with the following exceptions: FORMAT, START/STOP UNIT, and WRITE BUFFER with the mode set to any of the Download microcode options. If one of the exception commands is received, the device server shall abort the self-test, update the self-test log, and service the command as soon as possible but not longer than two seconds after the command descriptor block has been validated. Whenever there are commands other than the exception commands to service while the device server is performing a self-test in the background mode, the device server shall suspend the self-

test as soon as possible to service the command. Suspension of the self-test and beginning to service the command shall never take longer than two seconds.

If a device server is performing a self-test in the foreground mode and an error occurs during the test, the device server shall update the Self-test results log page and report CHECK CONDITION/HARDWARE ERROR/LOGICAL UNIT FAILED SELF-TEST. The application client may obtain additional information about the failure by reading the log page. If the device server is unable to update the Self-test results log page it shall report CHECK CONDITION/HARDWARE ERROR/LOGICAL UNIT UNABLE TO UPDATE SELF-TEST LOG.

The application client may obtain information about the twenty most recently completed self-tests by reading the Self-test results log page. This is the only method for an application client to obtain information about self-tests performed in the background mode.

An application client may terminate a self-test that is being performed in the foreground mode by sending an Abort Task message. An application client may terminate a self-test that is being performed in the background mode by issuing a SEND DIAGNOSTICS command with the Abort background self-test function.

The PROGRESS INDICATION field returned in response to a REQUEST SENSE command may be used by the application client at any time during execution of a self-test to poll the logical unit's progress.

**[EDITOR'S NOTE (mse – 26 Apr 99): Use of the RECEIVE DIAGNOSTIC RESULTS command to poll for progress was considered and was rejected in favor of REQUEST SENSE for at least three reasons: 1) it was thought that one RECEIVE DIAGNOSTIC RESULTS needed to be paired with one SEND DIAGNOSTICS command so that an application client would be unable to poll multiple times during a self-test; 2) the description of the SEND DIAGNOSTICS command in SPC-2 says that this command is usually followed by a RECEIVE DIAGNOSTIC RESULTS command *except following a self-test*; and, 3) it was thought that most devices would handle a REQUEST SENSE command more easily than a RECEIVE DIAGNOSTIC RESULTS command while performing a self-test in the foreground mode.]**

The Table 1 summarizes when a logical unit returns status after receipt of a self-test command, how an application client may abort a self-test, how a logical unit handles new commands that are received while a self-test is in progress, and how a logical unit reports a self-test failure.

**Table 1 – Self-test mode summary**

Mode	When Status is Returned	How to abort the self-test	Processing of subsequent commands while self-test is executing	Self-test failure reporting
Fore-ground	After the self-test is complete	Abort Task message	CHECK CONDITION / NOT READY / LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS [ <b>note 1</b> ] (except INQUIRY, REPORT LUNS and REQUEST SENSE)	CHECK CONDITION / HARDWARE ERROR / LOGICAL UNIT FAILED SELF-TEST [ <b>note 2</b> ] or / LOGICAL UNIT UNABLE TO UPDATE SELF-TEST LOG [ <b>note 3</b> ]
Back-ground	After the CDB for the self-test is validated	SEND DIAGNOSTICS with Abort background self-test function	Process the command (except FORMAT, START/STOP UNIT, and WRITE BUFFER with the mode set to any of the Download microcode options – then abort the self-test)	Application client checks Self-test results log page after the PROGRESS INDICATION field returned from REQUEST SENSE indicates the self-test is complete
<b>[EDITOR'S NOTE (mse – 23 Jul 99): (1) it is recommended that this code be 04h,09h; (2) it is recommended that this code be 3Eh,03h; (3) it is recommended that this code be 3Eh,04h.</b>				

[EDITOR'S NOTE (mse – 23 Jul 99): The following is from the SEND DIAGNOSTICS clause in SPC-2, "If the self-test successfully passes, the command shall be terminated with GOOD status; otherwise, the command shall be terminated with CHECK CONDITION status and the sense key shall be set to HARDWARE ERROR." This looks to me to be identical with the corresponding elements in the Foreground mode above. Unfortunately, there is no specification in SPC-2 about how to abort a self-test, or what is supposed to happen if a new command is received. I think that SPC-2 ought to be updated to include what we have identified here for those actions for SEND DIAGNOSTICS.]

#### 4 Mode page entry

The following describes the EXTENDED SELF-TEST ROUTINE COMPLETION TIME field:

[EDITOR'S NOTE (mse – 26 Apr 99): A mode page needs to be identified where this could be placed. I recommend that bytes 10 and 11 (currently "reserved") in the Control mode page be used for this (see 8.3.4 in SPC-2). The following assumes that this recommendation is accepted.]

Table 2 – Control mode page

Bit	7	6	5	4	3	2	1	0	
Byte									
....	....								
10	(MSB)	EXTENDED SELF-TEST ROUTINE COMPLETION TIME							
11							(LSB)		

The EXTENDED SELF-TEST ROUTINE COMPLETION TIME is an advisory parameter that the application client may use to determine the time in seconds that the device server requires to complete the extended self-test routine when the device server is not interrupted by an application client and no errors occur during execution of the self-test routine. The application client should expect this time to increase significantly if other commands are sent to the logical unit while a self-test is in progress or if errors occur during execution of the self-test.

Device servers supporting FUNCTION CODE field values other than 000b for the SEND DIAGNOSTIC command, shall support the EXTENDED SELF-TEST ROUTINE COMPLETION TIME field.

#### 5 Modifications to the SEND DIAGNOSTICS command

The last line in the first paragraph of this clause in SPC-2 is changed to, "Except when the SELFTEST bit is one or the FUNCTION CODE field is not 000b, this command is usually followed by a RECEIVE DIAGNOSTIC RESULTS command (see 7.15)."

[Editor's Note (mse – 26 Apr 99): Reserved bits 7, 6 and 5 in byte 1 of the SEND DIAGNOSTICS command are changed to a FUNCTION CODE field, and reserved bit 3 in byte 1 is changed to an IMMED bit. The following table and text describes the proposed changes for this command.]

Table 3 – SEND DIAGNOSTICS command

Bit	7	6	5	4	3	2	1	0	
Byte									
0	OPERATION CODE (1Dh)								
1	FUNCTION CODE			PF	RESERVED	SELFTEST	DEVOFFL	UNITOFFL	
2	RESERVED								
3	(MSB)	PARAMETER LIST LENGTH							
4							(LSB)		
5	CONTROL								

When the SELFTEST bit is set to one the FUNCTION CODE field shall be 000b. When the SELFTEST bit is cleared to zero the content of FUNCTION CODE field is specified in Table 4.

**Table 4 – Function code field values**

Value	Function Name	Description
000b	NA	Value to be used when the SELFTEST bit is set to one or if the SEND DIAGNOSTIC command is not invoking one of the other self-test function codes (e.g., Translate Address page (see SBC)).
001b	Background short self-test	The device server shall start its short self-test routine in the background mode.
010b	Background extended self-test	The device server shall start its extended self-test routine in the background mode.
011b	NA	Reserved
100b	Abort background self-test	Abort the current short or extended self-test routine running in background mode. This value is only valid if a previous SEND DIAGNOSTIC command specified a Background short self-test or Background extended self-test function and that function has not completed. If any of these conditions is not true, then the device server shall return CHECK CONDITION/ILLEGAL REQUEST/INVALID FIELD IN CDB.
101b	Foreground short self-test	The device server shall start its short self-test routine in the foreground mode.
110b	Foreground extended self-test	The device server shall start its extended self-test routine in the foreground mode.
111b		Reserved

**6 Self-test results log page**

This log page provides the results from the twenty most recent self-tests. Results from the most recent self-test or the self-test currently in progress shall be reported in the first self-test log structure parameter; results from the second most recent self-test shall be reported in the second self-test log structure parameter; etc. If fewer than twenty self-tests have occurred, the unused self-test log structure parameter entries shall be zero filled.

Table 5 describes the Self-test results log page format that shall be returned by the device server upon request by the application client.

**Table 5 – Self-test results log page format**

Bit	7	6	5	4	3	2	1	0	
<b>Byte</b>									
<b>0</b>	PAGE CODE (10h)								
<b>1</b>	RESERVED								
<b>2</b>	(MSB)	PAGE LENGTH (0190h)							
<b>3</b>								(LSB)	
SELF-TEST RESULTS LOG PARAMETERS									
<b>4</b>									
.....									
<b>23</b>									
.....									
<b>384</b>									
.....									
<b>403</b>									

The PAGE CODE field shall be 10h.

The PAGE LENGTH field specifying the length in bytes of the parameter list that shall be transferred from the device server to the application client shall be 0190h.

Table 6 describes the  $n^{\text{th}}$  SELF-TEST RESULTS LOG PARAMETER field.

**Table 6 –  $n^{\text{th}}$  self-test results log parameter**

Bit	7	6	5	4	3	2	1	0
<b>Byte</b>								
<b>0</b>	(MSB) _____ PARAMETER CODE _____ (LSB)							
<b>1</b>								
<b>2</b>	DU (= 0)	DS (= 0)	TSD (= 0)	ETC (= 0)	TMC (= 00b)		LBIN (= 1)	LP (= 1)
<b>3</b>	PARAMETER LENGTH (0Ch)							
<b>4</b>	RSVD	FUNCTION CODE			SELF-TEST RESULTS VALUE			
<b>5</b>	EXTENDED SEGMENT NUMBER							
<b>6</b>	(MSB) _____ TIMESTAMP _____ (LSB)							
<b>7</b>								
<b>8</b>	(MSB) _____							
<b>.....</b>	LBA OF FIRST FAILURE							
<b>15</b>	_____ (LSB)							

The PARAMETER CODE field identifies the log parameter being transferred for that log page. This shall be the number of the self-test run. This number shall increment for each new test and wrap when saturated. That is, when this value has reached FFFFh for a self-test it shall be 0000h for the next self-test.

The FUNCTION CODE field contains the content of the FUNCTION CODE field in the SEND DIAGNOSTICS command that initiated this self-test. See Table 3 for a description of the FUNCTION CODE field values.

Table 7 defines the content of the SELF-TEST RESULTS VALUE field.

**Table 7 – Self-test results values**

Value	Description
0h	The self-test routine completed without error.
1h	The self-test routine was aborted by the application client.
2h	The self-test routine was interrupted by a reset.
3h	An unknown error occurred while the device server was executing the self-test routine and the device server was unable to complete the self-test routine.
4h	The self-test completed having a test element that failed, and the test element that failed is not known.
5h	The first segment of the self-test failed.
6h	The second segment of the self-test failed.
7h	The third or greater segment of the self-test failed (see the EXTENDED SEGMENT NUMBER field).
8h-Eh	Reserved
Fh	The self-test is in progress

The EXTENDED SEGMENT NUMBER field may be used to identify the number of the segment that failed during the self-test. If no segment failed, this field shall contain 00h.

If the logical unit implements logical blocks, the content of the LBA OF FIRST FAILURE field is the first logical block address where a self-test error occurred. This implies nothing about the quality of any other LBA on

the logical unit. The testing during which the error occurred may not have been performed in a sequential manner. This value shall not change (e.g., as the result of block reassignment). The content of the LBA OF FIRST FAILURE field shall be FFFFFFFFFFFFFFFFh if no errors occurred during the self-test or if the error that occurred is not related to a logical block address.

The TIMESTAMP field contains the total accumulated power-on hours of the device server at the time the self-test operation was completed. If the test is still in progress, the content of the TIMESTAMP field shall be zero. If the power-on hours for the device server at the time the self-test operation was completed is greater than FFFFh then the content of the TIMESTAMP field shall be FFFFh.