

Date: July 14, 2005

To: T10 Committee (SCSI)

From: George Penokie (IBM/Tivoli)

Subject: SPC-4: Enable Background Operations Error Reporting Bit

## 1 Overview

SPC-3 currently defines a background operation but does not define any good way to report a failure. There is also a new proposal (04-198) which adds in more background operations. Neither method provides any way to report errors that occur during the scan. To find out if there have been any errors the application client must poll using the REQUEST SENSE command then on indication that the background operation is complete issue a LOG SENSE command to determine if there were any errors or poll just using the LOG SENSE command to find if there have been any errors.

The problem is that polling is not something applications want to have to do. It's a very inefficient way to operate.

This proposal removes the need to poll and replaces the polling with an indication that an error has occurred with a Unit Attention.

## 2 SPC-4 Changes

### 2.0.1 Informational Exceptions Control mode page

The Informational Exceptions Control mode page (see table 1) defines the methods used by the target to control the reporting and the operations of specific informational exception conditions. This page shall only apply to informational exceptions that report an [ADDITIONAL SENSE CODE field](#) of FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the application client. The mode page policy (see 6.7) for this mode page shall be shared, per initiator port, or per I\_T nexus.

Informational exception conditions occur as the result of [background scan errors](#), [background self-test errors](#), or vendor specific events within a target. An informational exception condition may occur asynchronous to any commands issued by an application client.

NOTE 1 - Storage devices that support SMART (Self-Monitoring Analysis and Reporting Technology) for predictive failure software should use informational exception conditions.

**Table 1 — Informational Exceptions Control mode page**

Bit Byte	7	6	5	4	3	2	1	0
0	PS	SPF (0b)	PAGE CODE (1Ch)					
1	PAGE LENGTH (0Ah)							
2	PERF	Reserved	EBF	EWASC	DEXCPT	TEST	<a href="#">EBACKERR</a>	LOGERR
3	Reserved				MRIE			
4	INTERVAL TIMER							
7								
8	REPORT COUNT							
11								

The PS bit, SPF bit, PAGE CODE field, and PAGE LENGTH field are described in 7.4.5.

The log errors (LOGERR) bit set to zero indicates that the logging of informational exception conditions by a device server is vendor specific. A LOGERR bit set to one indicates the device server shall log informational exception conditions.

A TEST bit set to one shall create a test device failure at the next interval time, as specified by the INTERVAL TIMER field, if the DEXCPT bit is set to zero. When the TEST bit is set to one, the MRIE and REPORT COUNT fields shall apply as if the TEST bit were zero. The test device failure shall be reported with an additional sense code of FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE). If both the TEST and the DEXCPT bits are one, the device server shall terminate the MODE SELECT command with CHECK CONDITION status. The sense key shall be set to ILLEGAL REQUEST with the additional sense code set to INVALID FIELD IN PARAMETER LIST. A TEST bit set to zero shall instruct the device server not to generate any test device failure notifications.

A disable exception control (DEXCPT) bit set to zero indicates the failure prediction threshold exceeded reporting shall be enabled. The method for reporting the failure prediction threshold exceeded when the DEXCPT bit is set to zero is determined from the mrrie field. A DEXCPT bit set to one indicates the target shall disable reporting of the failure prediction threshold exceeded. The mrrie field is ignored when DEXCPT is set to one and EWasc is set to zero.

An enable warning (EWasc) bit set to zero indicates the target shall disable reporting of the warning. The mrrie field is ignored when DEXCPT is set to one and EWasc is set to zero. An EWasc bit set to one indicates warning reporting shall be enabled. The method for reporting the warning when the EWasc bit is set to one is determined from the mrrie field.

[An enable background error \(EBACKERR\) bit set to zero indicates the target shall disable reporting of background self-test errors \(see 4.0.0.1\) and background scan errors \(see SBC-3\). An EBACKERR bit set to one indicates reporting of background self-test errors and background scan errors shall be enabled. The method for reporting background self-test errors and background scan errors is determined from the MRRIE field.](#)

### 3 New ASCQs

0Bh xxh DTLPWROMAEBKVF WARNING - BACKGROUND SELF-TEST FAILED

0Bh xxh DTLPWROMAEBKVF WARNING - BACKGROUND PRE-SCAN FAILED

0Bh xxh DTLPWROMAEBKVF WARNING - BACKGROUND MEDIUM SCAN FAILED

### 4 Additional SPC-4 Changes

#### 4.0.0.1 Features common to foreground and background self-test modes

The PROGRESS INDICATION field returned in response to a REQUEST SENSE command (see 6.27) may be used by the application client at any time during a self-test operation to poll the logical unit's progress. While a self-test operation is in progress unless an error has occurred, a device server shall respond to a REQUEST SENSE command by returning a sense key of NOT READY and an additional sense code of LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS with the sense key specific bytes set for progress indication.

[An application client may use the EBACKERR bit and the MRRIE field in the Information Exception Condition mode page \(see 2.0.1\) to report errors that occur during a background self-test operation.](#)

The application client may obtain information about the twenty most recently completed self-tests by reading the Self-Test Results log page (see 7.2.10). This is the only method for an application client to obtain information about self-tests performed in the background mode [unless an error occurs](#).

Table 2 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 2 — 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 task management function	If the command is INQUIRY, REPORT LUNS or REQUEST SENSE, process normally. Otherwise, terminate with CHECK CONDITION status, NOT READY sense key, and LOGICAL UNIT NOT READY, SELF-TEST IN PROGRESS additional sense code.	Terminate with CHECK CONDITION status, HARDWARE ERROR sense key, and LOGICAL UNIT FAILED SELF-TEST or LOGICAL UNIT UNABLE TO UPDATE SELF-TEST LOG additional sense code.
Back-ground	After the CDB is validated	SEND DIAGNOSTIC command with SELF-TEST CODE field set to 100b	Process the command, except as described in 5.5.3.2.	Application client: a) checks Self-Test Results log page (see 7.2.10) after the PROGRESS INDICATION field returned from REQUEST SENSE indicates the self-test is complete; <u>or</u> b) <u>uses the EBACKERR bit and the MRIE field (see 2.0.1) to select a method of indicating a failure occurred. If a failure occurs an additional sense code of WARNING - BACKGROUND SELF-TEST FAILED shall be returned using the method defined in the MRIE field.</u> <sup>a</sup>
<sup>a</sup> <u>The device server shall not report an error until after the Self-Test Results log page is updated.</u>				

## 5 Additions to proposal 04-198 (section 4.x.4 Interpreting the logged results)

An application client may:

a) poll the Background Scan Results log page (see 6.2.x) to get information about pre-scan and background medium scan activity; or

b) use the EBACKERR bit and the MRIE field (see 2.0.1) to select a method of indicating a failure occurred. If the EBACKERR bit is set to one and a failure occurs the following additional sense codes shall be returned using the method defined in the MRIE field:

- A) if the failure occurs during a pre-scan an additional sense code of WARNING - BACKGROUND PRE-SCAN FAILED shall be returned; or
- B) if the failure occurs during a scan an additional sense code of WARNING - BACKGROUND MEDIUM SCAN FAILED shall be returned.

**I** If an application client **may** polls the Background Scan Results log page (see 6.2.x) to get information about pre-scan and background medium scan activity, **then** the BMS status parameter has fields that **may be used to** indicate whether a pre-scan or background medium scan is active or suspended, the number of scans performed on the medium, and the progress of a scan that is active. This information may be used by an application client to monitor the background scanning operations.

[If an application client uses the EBACKERR bit and the MRIE field \(see 2.0.1\) to indicate a pre-scan or background medium scan failure and a failure occurs, then the application client should retrieve the Background Scan Results log page to get information about the failure.](#)

The medium scan parameters, if any, describe the physical and logical location of any suspected bad blocks. The REASSIGN STATUS field indicates whether the defect was completely handled by the device server or whether the application client may have to take action (e.g., reassigning or re-writing an LBA) to fix a particular bad block.

After an application client analyzes the medium scan parameters and has completed actions, if any, to resolve the bad blocks, it may delete the log entries by issuing a LOG SELECT command with PCR bit set to one (see Log Select command in SPC-3).