

To: T10 Technical Committee
 From: Rob Elliott, HP (elliott@hp.com)
 Date: 9 September 2005
 Subject: 05-340r0 SBC-3 SPC-4 Background scan additions

Revision history

Revision 0 (9 September 2005) First revision

Related documents

sbc2r16 - SCSI Block Commands - 2 revision 16
 spc4r01a - SCSI Primary Commands - 4 revision 1a
 04-198r5 - SBC-3 Background Medium Scan (Gerry Houlder, Seagate)
 04-317r1 - SPC-3 Enable Background Operations Error Reporting Bit (George Penokie, IBM)

Overview

Several changes and clarifications are proposed to the background scan operation recently added to SBC-3:

1. The additional sense code names proposed in 04-317r1 of WARNING - PRE-SCAN FAILED and WARNING - SCAN FAILED were interpreted by several readers as meaning the entire background scan failed, not just that it detected one or more bad logical blocks. These should be renamed to WARNING - PRE-SCAN DETECTED MEDIUM DEFECT and WARNING - BACKGROUND SCAN DETECTED MEDIUM DEFECT
2. There is no guidance for how long the logical unit should be idle before it starts performing the background scan. If software issues commands just as the logical unit chooses to resume a background scan, performance will degrade. A mode page field is proposed to both report and select the idle time that a logical unit will wait for before resuming scanning.
3. There is no guidance for how long the logical unit should keep attempting to reallocate a logical block detected as problematic during a background scan before it starts attending to a new command. A mode page field is proposed to both report and select the maximum time for stopping the background scan operation and resuming work on new commands.
4. It is unclear whether informational exceptions reported based on the EBACKERR field honor the INTERVAL TIMER field.
5. It is unclear whether informational exceptions reported based on the REPORT COUNT field. If they do, does this apply to reporting the same LBA, or faults detected during the same pass, or faults detected during all passes?
6. It is unclear whether the new EBACKERR field honors the INTERVAL TIMER and REPORT COUNT fields.
7. It is unclear whether the TOTAL POWER ON MINUTES field contains the total time the drive has been powered on since manufacturing, or the time since the last power on.
8. The resolution of minutes in the log page is rather large, making it hard to distinguish errors. Should the TOTAL POWER ON MINUTES field just be replaced with a simple sequence number instead? Could some of the vendor-specific bytes in the medium scan parameter into a sequence count, so more LBAs can be logged within a given timestamp?
9. A few other questions are scattered throughout as editor's notes

Suggested changes to SBC-3 as modified by 04-198r5 then 04-317r1

4.x.2 Background pre-scan feature

4.x.2.1 Enabling background pre-scan operation

The background pre-scan feature is enabled by setting the EN_PS bit in the Background Control mode page (see 6.3.x) to zero, then setting the EN_PS bit to one and power cycling the device. After power is restored the pre-scan timer is set to zero and the device [server](#) begins scanning the medium starting at LBA zero and ending with the last LBA. During this time any commands from an application client shall cause the [background](#) pre-scan operation to be suspended [within the time specified in the MAXIMUM TIME TO SUSPEND](#)

[BACKGROUND SCAN](#) field in the [Background Control mode page](#) (see 6.3.x) while the device server processes the commands. The [background](#) pre-scan operation shall resume where it left off when all commands have been completed (i.e., no ACA exists) [and the logical unit has been idle for the time specified in the IDLE TIME field in the Background Control mode page](#) (see 6.3.x).

Editor's Note 1: most other timers are described as counting down and expiring, rather than starting at zero and counting up.

A write operation that accesses an LBA that has not been scanned during the [background](#) pre-scan operation is converted into a write operation followed by a verify operation. This verifies that the data just written was read back successfully. If a write operation accesses an LBA that has already been scanned during the [background](#) pre-scan operation then it is processed normally. Commands that do not perform write operations are processed normally.

Editor's Note 2: consider providing a way to report the current LBA that is being pre-scanned, so software knows which writes are going to be slow. A unit attention on completion might be appropriate too.

4.x.2.2 Halting background pre-scan operation

The logical unit shall halt ~~F~~the background pre-scan ~~feature is halted~~ when any of the following occurs:

- a) the [background](#) pre-scan operation completes scanning all LBAs on the device;
- b) an ~~initiator~~ [application client](#) clears the EN_PS bit (see 6.3.x);
- c) the pre-scan timer reaches the Pre-scan Timeout Value (see 6.3.x); or
- d) the device [server](#) detects a vendor-specific pattern of errors.

Once the [background](#) pre-scan is halted, it is re-enabled as described in 4.x.2.1.

4.x.3 Background medium scan feature

4.x.3.1 Enabling background medium scan operation

If background pre-scan ~~operation~~ is enabled, it shall be completed or halted before [a](#) background medium scan operation is started. The background medium scan ~~feature~~ is enabled by setting the EN_BMS bit in the Background Control mode page (see 6.3.x). After the BMS Interval Time (see 6.3.x) has been reached [and the logical unit has been idle for the time specified in the IDLE TIME field in the Background Control mode page](#) (see 6.3.x), the device [server](#) shall begin scanning the medium starting at LBA zero and ending with the last LBA. During this scan any commands from an application client shall cause the background medium scan operation to be suspended while the device server processes the commands. The background medium scan operation shall resume where it left off when all ~~initiator~~ commands have been successfully completed (i.e., no ACA exists) [and the logical unit has been idle for the time specified in the IDLE TIME field in the Background Control mode page](#) (see 6.3.x).

After the last LBA has been scanned, the BMS interval timer is set to zero and the background medium scan operation becomes dormant for [at least a](#) BMS Interval Time. Then the background medium scan operation starts another scanning cycle at LBA zero.

Editor's Note 3: most other timers are described as counting down and expiring, rather than starting at zero and counting up.

4.x.3.2 Suspending background medium scan operation

The logical unit shall suspend ~~the~~ the background medium scan ~~feature is suspended~~ when any of the following occurs:

- a) the background medium scan operation completes scanning all LBAs on the device;
- b) an initiator sets the EN_BMS bit to zero (see 6.3.x); or
- c) the device detects a vendor specific pattern of errors.

The logical unit shall suspend within the time specified in the MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field in the Background Control mode page (see 6.3.x).

Once the background medium scan is suspended, it resumes from the ~~spot~~location where it left off when it is enabled (see 4.x.3.1).

Editor's Note 4: Does it always result from where it left off? If EN_BMS is set to zero, perhaps it goes back to LBA zero (which is how 4.x.3.1 reads).

Editor's Note 5: There are two kinds of "suspends" here - suspending temporarily to process commands and suspending (almost halting) due to completion, EN_BMS bit, or errors. Should they be described with different verbs?

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 [background](#) 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 [background](#) pre-scan an additional sense code of WARNING - [BACKGROUND PRE-SCAN FAILED DETECTED MEDIUM DEFECT](#) shall be returned; or
 - B) if the failure occurs during a [background medium](#) scan an additional sense code of WARNING - [BACKGROUND MEDIUM SCAN FAILED DETECTED MEDIUM DEFECT](#) shall be returned.

If an application client polls the Background Scan Results log page (see 6.2.x) to get information about [background](#) pre-scan and background medium scan activity, then the BMS status parameter has fields that may be used to indicate whether a [background](#) 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 an ~~failure detected during a background~~ [failure detected during a background](#) 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 [logical](#) 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 [logical](#) block.

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

Editor's Note 6: deleting is impolite in a multi-initiator environment. Would it be better to let it update the log entry with LOG SELECT to mark an LBA as "fixed by an application client" but

leave the LBA in the list?

Editor's Note 7: is it clear that clearing the log is atomic w.r.t. any scan currently occurring?

6.2.x.2 Medium scan parameter

A medium scan parameter (see table 1) describes a defect location on the medium that was encountered by background scanning (see 4.x.2).

Table 1 — Medium scan parameter format

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB) _____							
1	PARAMETER CODE (0001h through 0800h) _____ (LSB)							
2	DU	DS	TSD	ETC	TMC	LBIN		LP
3	PARAMETER LENGTH (14h)							
4	(MSB) _____							
7	POWER ON MINUTES _____ (LSB)							
	REASSIGN STATUS				SENSE KEY			
	ADDITIONAL SENSE CODE							
7	ADDITIONAL SENSE CODE QUALIFIER							
11	Vendor specific _____							
15	_____							
16	(MSB) _____							
23	LBA _____ (LSB)							

The POWER ON MINUTES [field](#) indicates the total power on minutes at the time the error was detected.

Editor's Note 8: Is this the total minutes powered on since manufacturing, or total minutes since power on?

Editor's Note 9: Minutes are a wide resolution, making it hard to tell errors apart and possibly overrunning 800h errors in a minute. The drive might not want to bother keeping track of time like this either. How about just making it a sequence number?

Table 2 ~~specifies the meaning of~~ defines the REASSIGN STATUS field.

Table 2 — ~~Reassign Status~~ REASSIGN STATUS field

<u>Value Code</u>	Description
0h	No reassignment needed.
1h	<u>The logical block specified by the LBA field failed and R</u> reassignment <u>is</u> pending receipt of Reassign: a) <u>a REASSIGN BLOCKS command (see 5.16); or</u> b) write command <u>a command performing a write operation.</u> (if auto write reallocation is allowed.) from the initiator.
2h	LBA <u>The logical block specified by the LBA field failed and was</u> successfully reassigned by drive <u>device server</u> .
3h	Reserved
4h	<u>The logical block specified by the LBA field failed and R</u> reassignment failed: <u>the logical block has an uncorrectable error</u>
5h	LBA <u>The logical block specified by the LBA field failed and was</u> recovered via re-write <u>an automatic reallocation by a write operation</u>
6h - Fh	Reserved

Editor's Note 10: When would this parameter be present with a code of 0h? Should that be marked reserved instead?

The SENSE KEY, ADDITIONAL SENSE CODE, and ADDITIONAL SENSE CODE QUALIFIER fields may contain a hierarchy of additional information relating to error conditions that occurred during background scanning. They are represented in the same format used by the sense data (see SPC-3).

The LBA field indicates the logical block address associated with the medium error.

6.3.x Background Control mode page

The Background Control mode page (see table 3) is a subpage of the Informational Exception Control mode page (see SPC-3) and provides controls over background operations. The mode page policy (see SPC-3) for this subpage shall be shared.

Table 3 — Background Control mode page

Byte/Bit	7	6	5	4	3	2	1	0
0	PS	SPF (1b)	PAGE CODE (1Ch)					
1	SUBPAGE CODE (01h)							
2	(MSB)	PAGE LENGTH (000Ch)						
3								(LSB)
4	Reserved							EN_BMS
5	Reserved							EN_PS
6	(MSB)	BMS INTERVAL TIME						
7								(LSB)
8	(MSB)	PRE-SCAN TIMEOUT VALUE						
9								(LSB)
10	(MSB)	IDLE TIME Reserved						
11								(LSB)
12	(MSB)	MAXIMUM TIME TO SUSPEND BACKGROUND SCAN Reserved						
13								(LSB)
14	Reserved							
15	Reserved							

An enable background medium scan (EN_BMS) bit set to zero specifies that background medium scan ~~is~~ shall be disabled. An EN_BMS bit set to one specifies that background medium scan operations ~~are~~ shall be enabled. If the EN_PS bit is also set to one then a background medium scan operation shall not start until after the ~~background~~ pre-scan operation is halted or completed. The BMS interval time shall occur before a background medium scan operation is started. If a background medium scan is in progress when the EN_BMS bit is changed from one to zero, then the ~~background~~ medium scan shall be suspended ~~before completing the MODE SELECT command and remain suspended~~ until the EN_BMS bit is set to one, at which time the ~~background~~ medium scan shall resume from the suspended location.

An enable pre-scan (EN_PS) bit set to zero specifies that ~~background~~ pre-scan ~~is~~ shall be disabled. If a ~~background~~ pre-scan operation is in progress when EN_PS is changed from a one to a zero then ~~the~~ ~~background~~ pre-scan operation shall be ~~is~~ halted ~~before completing the MODE SELECT command~~. An EN_PS bit set to one specifies that a ~~background~~ pre-scan operation ~~is~~ shall be started after the next power on ~~eyele~~. Once this ~~background~~ pre-scan has completed, another ~~background~~ pre-scan shall not occur unless the EN_PS bit is set to zero, then set to one, and another power on ~~eyele~~ occurs.

The BMS interval time field specifies the minimum time, in hours, between the end of one pre-scan or background medium scan operation and the start of the next background medium scan operation.

The pre-scan timeout value field specifies the maximum time, in hours, for a pre-scan operation to complete. If the pre-scan operation does not complete within the specified time then it is halted. A value of zero specifies an unlimited timeout value.

The IDLE TIME field specifies the time, in milliseconds, that the logical unit shall be idle before resuming a background pre-scan or a background medium scan.

[The MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field specifies the time, in milliseconds, that the logical unit shall take to start processing a command received while it is performing a background pre-scan or a background medium scan.](#)

Suggested changes to SPC-4

Global: Change WARNING - PRE-SCAN FAILED to WARNING - BACKGROUND PRE-SCAN DETECTED MEDIUM DEFECT

Global: Change WARNING - SCAN FAILED to WARNING - BACKGROUND MEDIUM SCAN DETECTED MEDIUM DEFECT

7.4.11 Informational Exceptions Control mode page

The Informational Exceptions Control mode page (see table 256) defines the methods used by the device server to control the reporting and the operations of specific informational exception conditions. This page shall only apply to informational exceptions that report additional sense codes with the ADDITIONAL SENSE CODE field set to FAILURE PREDICTION THRESHOLD EXCEEDED or WARNING to the application client. The mode page policy (see 6.7) for this mode page shall be shared, 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 logical unit. An informational exception condition may occur asynchronous to any commands issued by an application client.

NOTE 46 - Storage devices that support SMART (Self-Monitoring Analysis and Reporting Technology) for predictive failure software should use informational exception conditions.

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

If the log errors (LOGERR) bit is set to zero, the logging of informational exception conditions by a device server is vendor specific. If the LOGERR bit is set to one, 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 the additional sense code set to FAILURE PREDICTION THRESHOLD EXCEEDED (FALSE). If both the TEST bit and the DEXCPT bit are [set to one](#), the MODE SELECT 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 PARAMETER LIST. ~~Alf the~~ TEST bit ~~is~~ set to zero ~~shall instruct~~ the device server [shall](#) not ~~to~~ generate any test device failure notifications.

An enable background error (EBACKERR) bit set to zero indicates the target shall disable reporting of background self-test errors (see 5.5.3.4) 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 by contents of the MRIE field.

[Editor's Note 11: Do background errors honor the INTERVAL TIMER field?](#)

[Editor's Note 12: Do background errors honor the REPORT COUNT field?](#)

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 MRIE field. A DEXCPT bit set to one indicates the device server shall disable reporting of the failure prediction threshold exceeded. The MRIE field is ignored when DEXCPT is set to one and EWASC is set to zero.

If the enable warning (EWASC) bit is set to zero, the device server shall disable reporting of the warning. The MRIE field is ignored when DEXCPT is set to one and [the EWASC bit](#) is set to zero. If the EWASC bit is set to one, warning reporting shall be enabled. The method for reporting the warning when the EWASCbit is set to one is determined from the MRIE field.

If background functions are supported and the ~~Enable-Background-Function~~[enable background function](#) (EBF) bit is set to one, then the device server shall enable background functions. If the EBF bit is set to zero, the device server shall disable the functions.

Background functions with separate enable control bits (e.g., background ~~medium read~~ scan defined in SBC-3) are not controlled by this bit.

For the purposes of the EBF bit, background functions are defined as idle time functions that may impact performance that are performed by a device server operating without errors but do not impact the reliability of the logical unit.

If the performance (PERF) bit is set to zero, informational exception operations that are the cause of delays are acceptable. If the PERF bit is set to one, the device server shall not cause delays while doing informational exception operations. A PERF bit set to one may cause the device server to disable some or all of the informational exceptions operations, thereby limiting the reporting of informational exception conditions.

The value in the method of reporting informational exceptions (MRIE) field defines the method that shall be used by the device server to report informational exception conditions (see table 257). The priority of reporting multiple information exceptions is vendor specific.

...

The value in the INTERVAL TIMER field is the period in 100 millisecond increments for reporting that an informational exception condition has occurred. The device server shall not report informational exception conditions more frequently than the time specified by the INTERVAL TIMER field and shall report them after the time specified by [the](#) INTERVAL TIMER field has elapsed. After the informational exception condition has been reported the interval timer shall be restarted. A value of zero or FFFF FFFFh in the INTERVAL TIMER field indicates that the period for reporting an informational exception condition is vendor specific.

The value in the REPORT COUNT field is the number of times to report an informational exception condition to the application client. A value of zero in the REPORT COUNT field indicates there is no limit on the number of times the device server reports an informational exception condition.

The maintaining of the interval timer and the report counter across power cycles, hard resets, logical unit resets, and I_T nexus losses is vendor specific.