

To: T10 Technical Committee
 From: Rob Elliott, HP (elliott@hp.com)
 Date: 11 November 2005
 Subject: 05-340r2 SBC-3 SPC-4 Background scan additions

Revision history

Revision 0 (9 September 2005) First revision

Revision 1 (18 October 2005) Incorporated comments from September 2005 CAP WG. Upgraded text to be based on 05-369r0 (proposed SBC-3 revision 2) and added some corrections to some incorporation errors of the earlier proposals into 05-369.

Revision 2 (11 November 2005) Incorporated comments from November 2005 CAP WG.

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)

05-344r0 - Proposed SBC-3 revision 0 (George Penokie, IBM) - based on sbc2r16

05-345r0 - Proposed SBC-3 revision 1 (George Penokie, IBM) - incorporates 04-198r5 and 04-317r1

05-369r0 - Proposed SBC-3 revision 2 (George Penokie, IBM)

05-375r0 - SBC-3 EBACKERR bit Error Reporting Clarification (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 ERROR and WARNING - BACKGROUND SCAN DETECTED MEDIUM ERROR
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 MINIMUM IDLE TIME BEFORE BACKGROUND SCAN 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 MAXIMUM TIME TO SUSPEND BACKGROUND SCAN 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 the new EBACKERR field honors the INTERVAL TIMER and REPORT COUNT fields. 05-375r0 resolves this (it does not honor them), so revision 1 of this proposal no longer discusses the issue.
5. 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. The WG recommended that it contain the total time the drive has been powered on since manufacturing.
6. Several paragraphs say that the scan starts at LBA zero and ends at the last LBA, implying a sequential scan. The WG recommended that the device server be allowed to scan any way it chooses (e.g., based on physical block layout rather than logical block layout) and not be required to scan in LBA order.
7. Background scanning should stop if the Background Scan Results log page becomes full of errors. Revision 1 proposed that this happen if the RLEC bit in the Control mode page is set to one (which generates a CHECK CONDITION/RECOVERED ERROR/LOG LIST CODES EXHAUSTED). Revision 2 proposes a new STOP ON LOG FULL bit instead.
8. Background medium scan should start immediately when EN_BMS is set to one, not wait a whole Interval time.
9. The maximum log page parameter code is inconsistent - should it be 0800h or 07FFh? 0800h is proposed.

10. A log parameter with a REASSIGN STATUS field set to 00h should never appear, so 00h should be marked Reserved.
11. The log parameters need to be defined as binary list parameters.
12. For consistency with other timer names, the PRE-SCAN TIMEOUT VALUE field is renamed BACKGROUND PRE-SCAN TIME LIMIT and its timer is called the Background Pre-Scan Timeout timer. The BMS INTERVAL TIME field is renamed BACKGROUND MEDIUM SCAN INTERVAL TIME and its timer is called the Background Medium Scan Interval timer.
13. For consistency with other timers, timer descriptions are changed to initializing to the value in the mode page, starting, and expiring (rather than initializing to 0 and counting up to the value in the mode page).
14. "Suspending" is the term used for temporarily stopping the scan to process commands, while "halting" is used for completing background pre-scan or completing a pass of background medium scan.
15. In SPC-4, the Self-Test Results log page includes a TIMESTAMP field that is basically the same as the Background Scan Results log page TOTAL POWER ON MINUTES field (except for the selected units of hours vs. minutes), so should have the same name. "Timestamp" is a term now taken by the SET/REPORT TIMESTAMP commands, so should not be used by the log pages for a different purpose.
16. The LBA field should be spelled out as LOGICAL BLOCK ADDRESS to follow SBC-2 conventions.
17. A bit is proposed to limit logging to medium errors demanding application client attention (ignoring those that the device server fixes on its own).

Suggested changes to proposed SBC-3 revision 2 (05-369r0)

4.18 Background scanning operations

4.18.1 Background scanning overview

During background scanning, ~~Medium scanning occurs when a~~the device server, ~~without using any bandwidth on the service delivery subsystem,~~ reads logical blocks from the medium for the purpose of:

- a) identifying logical blocks that are difficult to read or unreadable;
- b) logging ~~a~~ read problems; and
- c) when allowed, take a vendor-specific action to make the logical block readable again.

~~Background medium scanning is defined as any operation that is performed without using any bandwidth on the service delivery subsystem (see SAM-4). Logical blocks shall not be retained in cache after they are read.~~

Editor's Note 1: Move first sentence up, second sentence down. Downgrade second sentence to a should. Use "cache memory" not "cache" for consistency with rest of SBC-2.

If a logical block is readable but requires extra actions (e.g., retries or application of a correction algorithm) to be read, the device server may resolve the problem using vendor-specific means. The ARRE bit in the Read-Write Error Recovery mode page (see 6.3.5) controls whether the device server may automatically repair or relocate recoverable read errors.

If a logical block is unreadable the device server may mark the logical block as bad so it may be relocated. The AWRE bit in the Read-Write Error Recovery mode page (see 6.3.5) controls whether the device server may relocate logical blocks during write operations. If allowed by the AWRE bits setting, logical blocks that have previously been noted as unrecoverable are reassigned at the start of the next write operation to that logical block.

During a background scan, the device server may scan the logical blocks in any order (e.g., based on physical block layout). The device server should not retain any logical blocks in cache memory after they are read.

4.18.2 Background pre-scan ~~feature~~

4.18.2.1 Enabling background pre-scan operation

The background pre-scan **feature** is enabled after:

- 1) the EN_PS bit in the Background Control mode page (see 6.3.3) is set to zero;
- 2) the EN_PS bit is set to one; and
- 3) ~~the SCSI device is power cycled~~ a power on occurs.

After power ~~is restored on,~~ the device server shall initialize the Background Pre-scan Time Limit timer to the time specified in the PRE-SCAN TIME LIMIT field specified in the Background Control mode page, start the Background Pre-scan Time Limit timer, and begin the background pre-scan operation (i.e., ~~pre-scan timer is set to zero and the SCSI device~~ begins scanning the medium) ~~starting at LBA zero and ending with the last LBA.~~

During ~~this time~~ the background pre-scan operation, any commands from an application client shall cause the background pre-scan operation to be suspended while the device server processes the commands. The background pre-scan operation should be suspended within the time specified in the MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field in the Background Control mode page (see 6.3.3). 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 MINIMUM IDLE TIME BEFORE BACKGROUND SCAN field in the Background Control mode page (see 6.3.3).

The device server shall convert each A write operation that accesses an LBAa logical block that has not been scanned during the background pre-scan operation is converted into a write operation followed by a verify operation to verify that the data just written was read back successfully. If a write operation accesses an LBAa logical block that has already been scanned during the background pre-scan operation then it is shall be processed normally. Commands that do not perform write operations are shall be processed normally.

4.18.2.2 Halting background pre-scan operation

The device server 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 logical blocks on the SCSI device medium;
- b) an application client sets the EN_PS bit to zero (see 6.3.3);
- c) the Background Pre-scan Time Limit timer ~~pre-scan timer expires~~ reaches the value of the PRE-SCAN-TIMEOUT-VALUE field (see 6.3.3);
- d) the Background Scanning Results log page Background Medium Scan log parameters are all used and the STOP_ON_LOG_FULL bit in the Background Control mode page (see 6.3.3) is set to one; or
- e) the SCSI device server detects a vendor-specific pattern of errors.

Once the background pre-scan is halted, it is re-enabled as described in 4.18.2.1.

4.18.3 Background medium scan **feature**

4.18.3.1 Enabling background medium scan operation

If background pre-scan **operation** (see 4.18.2) 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 is set to one in the Background Control mode page (see 6.3.3) to one.

The device server shall begin a background medium scan operation (i.e. begin scanning the medium) when:

- a) background medium scan is enabled;
- b) if the device server has previously performed a background medium scan operation since power on, the Background Medium Scan Interval timer has expired; and
- c) the logical unit has been idle for the time specified in the MINIMUM IDLE TIME BEFORE BACKGROUND SCAN field in the Background Control mode page (see 6.3.3)

~~After the value in the BMS INTERVAL TIME field (see 6.3.3) has been reached, the device shall begin scanning the medium starting at LBA zero and ending with the last LBA.~~ During ~~this~~ the background medium scan operation, 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 should be suspended within the time specified in the MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field in the

Background Control mode page (see 6.3.3). The background medium scan operation shall resume where it left off when all commands have been successfully completed ~~(i.e., no ACA exists)~~, and the logical unit has been idle for the time specified in the MINIMUM IDLE TIME BEFORE BACKGROUND SCAN field in the Background Control mode page (see 6.3.3).

Editor's Note 2: i.e. is wrong, "no ACA exists" does not restate "all commands have been successfully completed"

After the ~~last LBA~~ entire medium has been scanned, the ~~BMS interval~~ Background Medium Scan Interval timer shall be initialized to the time specified in the BACKGROUND MEDIUM SCAN INTERVAL TIME field in the Background Control mode page and started. ~~shall be set to zero and the background medium scan operation becomes dormant for a BMS interval time. Then the background medium scan operation starts another scanning cycle at LBA zero.~~ After the Background Medium Scan Interval timer expires, the device server shall start another background medium scan operation.

Editor's Note 3: incorporation of the proposal was incorrect - the timer field is not set to zero. The timer itself is set to zero.

4.18.3.2 ~~Suspending~~ **Halting** background medium scan operation

The device server shall halt ~~T~~the background medium scan operation ~~feature is suspended~~ when any of the following occurs:

- a) the background medium scan operation completes scanning all ~~LBAs~~ logical blocks on the ~~device~~ medium;
- b) an application client sets the EN_BMS bit to zero (see 6.3.3);
- c) the Background Scanning Results log page Background Medium Scan log parameters are all used and the STOP_ON_LOG_FULL bit in the Background Control mode page (see 6.3.3) is set to one; or
- d) the ~~SGS~~ device server detects a vendor-specific pattern of errors.

The device server should halt the background medium scan operation within the time specified in the MAXIMUM TIME TO SUSPEND BACKGROUND SCAN field in the Background Control mode page (see 6.3.3).

Once the background medium scan is ~~suspended~~ halted, it resumes from the ~~spot~~ location where it left off when it is enabled (see 4.18.3.1).

4.18.4 Interpreting the logged results

An application client may:

- a) poll the Background Scan Results log page (see 6.2.2) to get information about background pre-scan and background medium scan activity; or
- b) use the EBACKERR bit and the MRIE field in the Informational Exceptions Control mode page (see SPC-4) to select a method of indicating ~~a failure occurred~~ that a medium error was detected. If the EBACKERR bit is set to one and ~~a failure occurs~~ a medium error is detected, the following additional sense codes shall be returned using the method defined in the MRIE field:
 - A) WARNING - BACKGROUND PRE-SCAN DETECTED MEDIUM ERROR if the failure occurs during a background pre-scan ~~an additional sense code of WARNING - PRE-SCAN FAILED shall be returned;~~ ~~or~~ and
 - B) WARNING - BACKGROUND MEDIUM SCAN DETECTED MEDIUM ERROR if the failure occurs during a background medium scan ~~an additional sense code of WARNING - SCAN FAILED shall be returned.~~

~~If an application client polls t~~ The Background Scan Results log page (see 6.2.2) to get information about background pre-scan and background medium scan activity, then the b Background s Status parameter (see table 92) ~~has fields that may be used to indicate~~ s whether a background pre-scan or background medium scan is active or ~~suspend~~ halted, the number of background scans performed on the

medium, and the progress of a [background](#) scan that is active. This information may be used by an application client to monitor the background scanning operations [and should be used by an application client after notification via an informational exception](#).

~~If an application client uses the EBACKERR bit and the MRIE field in the Informational Exceptions Control mode page (see SPC-4) to indicate an 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 Background [mMedium](#) [sScan](#) parameters (see table 94), if any, describe the ~~physical and~~ logical location of any suspected bad [logical](#) blocks. The REASSIGN STATUS field (see table 95) indicates whether the defect was completely handled by the device server or whether the application client ~~may have needs~~ to take action (e.g., reassigning or re-writing ~~an LBA a~~ [logical block](#)) to fix a particular [bad](#) logical block.

After an application client analyzes the Background [mMedium](#) [sScan](#) parameters and has completed actions, if any, to repair the [bad](#) logical blocks, it may delete the log entries by issuing a LOG SELECT command ([e.g., with the PCR bit set to one, or with the PC bit set to 11b and the PARAMETER LIST LENGTH field set to zero](#)) (see SPC-34).

[The background medium scan continues to run during log page accesses. To ensure that the log page does not change during a sequence of accesses, the application client shall:](#)

- 1) [set the EN_BMS bit in the Background Control mode page \(see 6.3.3\) to zero to halt the background medium scan;](#)
- 2) [read the log page with LOG SENSE command;](#)
- 3) [process the log page;](#)
- 4) [delete the log entries with the LOG SELECT command \(e.g., with the PCR bit set to one\); and](#)
- 5) [set the EN_BMS bit in the Background Control mode page \(see 6.3.3\) to one.](#)

5.16 REASSIGN BLOCKS command

5.16.1 REASSIGN BLOCKS command overview

The REASSIGN BLOCKS command (see table 44) requests that the device server reassign defective logical blocks to another area on the medium set aside for this purpose. The device server should also record the location of the defective logical blocks in the GLIST, if supported. This command shall not alter the contents of the PLIST (see 4.8).

The parameter list provided in the data-out buffer contains a defective LBA list that contains the LBAs of the logical blocks to be reassigned. The device server shall [either](#) reassign the parts of the medium used for each logical block in the defective LBA list [or rewrite the logical block in-place](#). More than one physical block may be relocated by each LBA. If the device server is able to recover user data and protection information, if any, from the original logical block, it shall write the recovered user data and any protection information to the reassigned [or rewritten](#) logical block. If the device server is unable to recover user data and protection information, if any, it shall write vendor-specific data as the user data and shall write a default value of FFFFFFFF_FFFFFFFFh as the protection information, if enabled. The data in all other logical blocks on the medium shall be preserved.

[Editor's Note 4: In the November CAP WG, disk drive vendors stated they might just rewrite, not always reallocate.](#)

6.2.2 Background Scan Results log page

The Background Scan Results log page (see table 90) returns the [bBackground](#) [sScanning](#) [sStatus](#) parameter and zero or more Background [mMedium](#) [sScan](#) parameters when background scanning is supported. The [bBackground](#) [sScanning](#) [sStatus](#) parameter provides information about background pre-scan and background medium scan operations. Each ~~medium error log entry~~ [Background Medium Scan parameter](#) corresponds to a logical block where an error was detected. If the ~~medium scan log area~~ [Background Scan Results log page](#) is filled up, a new ~~medium scan log entry~~ [Background Medium Scan parameter](#) overwrites the oldest entry. When a LOG SELECT command with the PCR bit set to one is processed all Background [mMedium](#) [sScan](#)

parameters are deleted; however, the values in the Background Scanning Status parameter shall not be affected.

Table 1 — Background Scan Results log page

Byte\Bit	7	6	5	4	3	2	1	0
0	Reserved		PAGE CODE (15h)					
1	Reserved							
2	(MSB)	PAGE LENGTH (n - 3)						(LSB)
3								
Background Scanning Results log parameters								
4	Background Scanning Status parameters (see table 3)							
19								
Background Medium Scan parameter list								
20	(MSB)	First Background Medium Scan parameter (first)(see table 5)						(LSB)
43								
...								
n-23	(MSB)	Last Background Medium Scan parameter (last)(see table 5)						(LSB)
n								

Table 91 defines the parameter codes for the Background Scan Results log page.

Table 2 — Background Scan Results log page parameter codes

Parameter code	Description
0000h	Background Scanning Status
0001h - 0800h	Background Medium Scan
07FFh-0801h - 7FFFh-FFFFh	Reserved

The Background Scanning Status parameter (see table 92) contains status information about the background pre-scan and background medium scan features.

Table 3 — Background Scanning Status parameter format

Byte\Bit	7	6	5	4	3	2	1	0
0	(MSB) _____							
1	PARAMETER CODE (0000h) _____ (LSB)							
2	DU	DS	TSD	ETC	TMC		LBIN	LP
3	PARAMETER LENGTH (0Ch)							
4	(MSB) _____							
7	ACCUMULATED POWER ON MINUTES _____ (LSB)							
8	Reserved							
9	BACKGROUND SCANNING STATUS							
10	(MSB) _____							
11	NUMBER OF BACKGROUND SCANS PERFORMED _____ (LSB)							
12	(MSB) _____							
13	BACKGROUND MEDIUM SCAN PROGRESS _____ (LSB)							
14	Reserved							
15	_____							

~~The contents of the DU, DS, TSD, ETC, LBIN, and LP bits and the TMC field are defined in SPC-4.~~

Table 4 defines the values for the log parameter control bits for this log parameter.

Table 4 — Parameter control bits for Background Scanning Status log parameter

Field	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0	any	The DU bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
DS	any	any	No specific requirements
TSD	any	any	No specific requirements
ETC	0	any	The ETC bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
TMC	00b	any	The TMC field is not defined for list parameters, so shall be set to 00b when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
LBIN	1	1	The log parameter is in binary format.
LP	1	1	The log parameter is a list parameter.

The PARAMETER LENGTH field indicates the number of bytes remaining in the log parameter.

The [ACCUMULATED POWER ON MINUTES](#) field indicates the ~~total power on minutes~~ [number of minutes the device server has been powered on since manufacturing](#) at the time the log page is requested.

Table 93 ~~specifies the meaning of~~ [defines](#) the BACKGROUND SCANNING STATUS field.

Table 5 — BACKGROUND SCANNING STATUS field

Value Code	Description
00h	No reassignment needed .
01h	Background medium scan is active
02h	Background P pre-scan is active
03h	Background s Scan halted due to fatal error
04h	Background s Scan halted due to a vendor-specific pattern of errors
05h	Background s Scan halted due to medium formatted without P-list
06h	Background s Scan halted - vendor-specific cause
07h	Background s Scan halted due to temperature out of range
08h	Background sScan suspended until BMS interval time (see 6.3.3) expires halted, waiting for Background Medium Scan Interval timer expiration
09h - FFh	Reserved

The NUMBER OF [BACKGROUND](#) SCANS PERFORMED field indicates the number of background scans that have been performed since the SCSI target device was originally shipped by the manufacturer.

The [BACKGROUND MEDIUM SCAN PROGRESS](#) field is a percent complete indication of the [background](#) medium scan. The returned value is a numerator that has 65 536 (i.e., 10000h) as its denominator.

A [Background m](#)Medium [s](#)Scan parameter (see table 94) describes a defect location on the medium that was encountered by background scanning (see 4.18.2).

Table 6 — [Background](#) Medium-~~s~~Scan parameter format

Byte/Bit	7	6	5	4	3	2	1	0
0	(MSB) _____							
1	PARAMETER CODE (0001h - 0800h) _____ (LSB)							
2	DU	DS	TSD	ETC	TMC		LBIN	LP
3	PARAMETER LENGTH (14h)							
4	(MSB) _____							
7	ACCUMULATED POWER ON MINUTES _____ (LSB)							
8	REASSIGN STATUS				SENSE KEY			
9	ADDITIONAL SENSE CODE							
10	ADDITIONAL SENSE CODE QUALIFIER							
11	_____							
15	VENDOR SPECIFIC Vendor specific _____							
16	(MSB) _____							
23	L B L O G I C A L L O G I C B L O C K A D D R E S _____ (LSB)							

The contents of the DU, DS, TSD, ETC, LBIN, and LP bits and the TMC field are defined in SPC-4.

Table 4 defines the values for the log parameter control bits (see SPC-4) for this log parameter.

Table 7 — Parameter control bits for Background Medium Scan log parameter

Field	Value for LOG SENSE	Value for LOG SELECT	Description
DU	0	any	The DU bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
DS	any	any	No specific requirements
TSD	any	any	No specific requirements
ETC	0	any	The ETC bit is not defined for list parameters, so shall be set to zero when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
TMC	00b	any	The TMC field is not defined for list parameters, so shall be set to 00b when read with the LOG SENSE command and shall be ignored when written with the LOG SELECT command.
LBIN	1	1	The log parameter is in binary format.
LP	1	1	The log parameter is a list parameter.

The PARAMETER LENGTH field indicates the number of bytes remaining in the log parameter.

The ACCUMULATED POWER ON MINUTES field indicates the ~~total power on minutes~~ number of minutes the device server has been powered on since manufacturing at the time the error was detected.

Table 8 ~~specifies~~ defines the REASSIGN STATUS field.

Table 8 — REASSIGN STATUS field

Value Code	Reported if the LOG_APP_ONLY bit is set to one in the Background Control mode page (see 6.3.3)	Description
0h	No	no reassignment needed. Reserved
1h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and Reassignment is pending receipt of Reassign: <u>a</u> <u>b</u> a) write command a command performing a write operation, (if auto write reallocation is allowed; or b) a REASSIGN BLOCKS command (see 5.16).
2h	No	LBA The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was successfully reassigned by the device server
3h	Reserved	
4h	No	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and Reassignment by the device server failed; the logical block has an uncorrectable error

Table 8 — REASSIGN STATUS field

Value Code	Reported if the LOG_APP_ONLY bit is set to one in the Background Control mode page (see 6.3.3)	Description
5h	No	LBA The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was recovered by the device server via re-write ^a rewrite in-place
6h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was successfully reassigned by the application client using a REASSIGN BLOCKS command ^b
7h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was successfully reassigned by the application client using a command performing a write operation ^b
8h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was successfully rewritten in-place by the application client using a REASSIGN BLOCKS command ^b
9h	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and was successfully rewritten in-place by the application client using a command performing a write operation ^b
Ah	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and failed reassignment by the application client using a REASSIGN BLOCKS command ^b
Bh	Yes	The logical block specified by the LOGICAL BLOCK ADDRESS field failed and failed reassignment by the application client using a command performing a write operation ^b
6hCh - Fh	Reserved	
<p>^a The application client should use a command performing a write operation if it knows what data belongs on the logical block (e.g., in a redundancy group (see 4.14.1), it uses data regenerated from the data on the other logical units in the redundancy group). The REASSIGN BLOCKS may not be able to recover the data and does not report whether or not it successfully does so.</p> <p>^b The REASSIGN STATUS field in a given log parameter changes from 1h to 6h, 7h, 8h, 9h, Ah, or Bh when the logical block is reassigned, rewritten, or failed. If the logical block is reassigned or rewritten, any subsequent medium error to the logical block is reported in a new log parameter with the same value in the LOGICAL BLOCK ADDRESS field.</p>		

The SENSE KEY field, ADDITIONAL SENSE CODE field, and the ADDITIONAL SENSE CODE QUALIFIER field 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-4).

The ~~LBA~~LOGICAL BLOCK ADDRESS field indicates the logical block address associated with the medium error.

6.3.3 Background Control mode page

The Background Control mode page (see table 9) is a subpage of the Informational Exception Control mode page (see SPC-4) and provides controls over background operations. The mode page policy (see SPC-34) for this subpage shall be shared.

Table 9 — 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					STOP_ON_LOG_FULL	LOG_APP_ONLY	EN_BMS
5	Reserved							EN_PS
6	(MSB)	BACKGROUND_MEDIUM_SCAN_BMS INTERVAL TIME						(LSB)
7								
8	(MSB)	BACKGROUND_PRE-SCAN_TIMEOUT_VALUE TIME LIMIT						(LSB)
9								
10	(MSB)	MINIMUM_IDLE_TIME_BEFORE_BACKGROUND_SCAN Reserved						(LSB)
11								
12	(MSB)	MAXIMUM_TIME_TO_SUSPEND_BACKGROUND_SCAN Reserved						(LSB)
13								
14	Reserved							
15								

Editor's Note 5: Delete vertical bars on the Reserved bytes at the end of the table

A [STOP_ON_LOG_FULL](#) field set to zero specifies that the device server shall continue running a background scan even if the Background Scan Results log page contains the maximum number of log parameters supported by the logical unit. A [STOP_ON_LOG_FULL](#) field set to one specifies that the device server shall halt a background scan if the Background Scan Results log page contains the maximum number of log parameters supported by the logical unit.

A [LOG_APP_ONLY](#) field set to zero specifies that the device server shall log all suspected medium errors in the Background Scan Results log page (see 6.2.2). A [LOG_APP_ONLY](#) field set to one specifies that the device server shall only log medium errors requiring application client intervention in the Background Scan Results log page as defined in table 8 in 6.2.2.

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 [BACKGROUND MEDIUM SCAN BMS](#) INTERVAL TIME field specifies the minimum time, in hours, between the end of one [background](#) pre-scan or background medium scan operation and the start of the next background medium scan operation. If the current background medium scan operation takes longer than the value specified in the [BACKGROUND MEDIUM SCAN BMS](#) INTERVAL TIME field, then the current background ~~pre-scan or background~~ medium scan ~~or pre-scan~~ continues until completion and the next background medium scan operation starts on completion of the current background ~~pre-scan or background~~ medium scan ~~or pre-scan~~.

The [BACKGROUND PRE-SCAN TIMEOUT VALUE TIME LIMIT](#) field specifies the maximum time, in hours, for a [background](#) pre-scan operation to complete. If the [background](#) pre-scan operation does not complete within the specified time then it is halted. A value of zero specifies an unlimited timeout value.

[The MINIMUM IDLE TIME BEFORE BACKGROUND SCAN](#) 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 should 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 ERROR](#)

[Global: Change WARNING - SCAN FAILED to WARNING - BACKGROUND MEDIUM SCAN DETECTED MEDIUM ERROR](#)

7.2.10 Self-Test Results log page

...

The [TIMESTAMP ACCUMULATED POWER ON HOURS](#) field contains the total ~~accumulated power on~~ [number of](#) hours ~~for~~ the device server [has been powered on since manufacturing](#) at the time the self-test was completed. If the test is still in progress, the ~~content of the~~ [TIMESTAMP ACCUMULATED POWER ON HOURS](#) field shall be [set to](#) zero. If the ~~power on hours for the device server at the time the self-test was completed~~ [number of hours](#) is greater than FFFFh then the ~~content of the~~ [TIMESTAMP ACCUMULATED POWER ON HOURS](#) field shall be [set to](#) FFFFh.

...

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. [If 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.

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.