



LEGAL NOTICE:

© **Copyright 2007 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED.**

This NVM Express revision 1.3 technical proposal is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this NVM Express revision 1.3 technical proposal subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: "© 2007 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED." When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

THIS DOCUMENT AND THE INFORMATION CONTAINED HEREIN IS PROVIDED ON AN "AS IS" BASIS. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, NVM EXPRESS, INC. (ALONG WITH THE CONTRIBUTORS TO THIS DOCUMENT) HEREBY DISCLAIM ALL REPRESENTATIONS, WARRANTIES AND/OR COVENANTS, EITHER EXPRESS OR IMPLIED, STATUTORY OR AT COMMON LAW, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE, VALIDITY, AND/OR NONINFRINGEMENT.

All product names, trademarks, registered trademarks, and/or servicemarks may be claimed as the property of their respective owners.

NVM Express Workgroup
c/o VTM Group
3855 SW 153rd Drive
Beaverton, OR 97003 USA
info@nvmexpress.org

NVM Express Technical Proposal for New Feature

Technical Proposal ID	4016 – Rebuild Assist
Change Date	09/17/2018
Builds on Specification	NVM Express 1.3a

Technical Proposal Author(s)

Name	Company
Bill Martin	Samsung
Judy Brock	Samsung
Fred Knight	NetApp
David Black	Dell EMC
Curtis Ballard	HPE

This technical proposal defines a new Get LBA Status capability that returns the status of LBAs on the device. This status may be used to determine what LBAs on a device should be recovered from another location and re-written. There are other potential uses for this functionality.

Revision History

Revision Date	Change Description
11/29/2017	Initial proposal
12/12/2017	Added support for returning list of currently known potentially-unrecoverable blocks, Collapsed the two NUMDL and NUMDU 16 bit fields into a single 32 bit NMDW field.
01/06/2018	Incorporate editorial feedback from Mike Allison and Judy Brock
01/06/2018	Add initial support for AEN and new log page
01/19/2018	Editorial changes and incorporate primarily editorial feedback from Bill Martin.
01/30/2018	<p>Get LBA Status now namespace-based. Modify LBA Status Information log page to return an array of namespace-scoped elements Add LBA Status Information Coalescing Feature Identifier Add support for AEN aggregation Add Generation Counter Add new 8.x Get LBA Status Capability section Added various new Identify Controller fields: Get LBA Status capability in OACS field, LBA Status Information Notice support in OAES field, default values for min aggregation time, default value for aggregation threshold Added status bit for logical blocks written by Write Unrecoverable command</p>
02/10/2018	<ul style="list-style-type: none"> Developed section 8.NEW text (new Get LBA Status capability description) Described relationship between LBA Range Descriptors returned by log page and LBA Status Descriptors returned by Get LBA Status command; provided examples Clarify in multiple places that the scan associated with Action Type 10h is an internal scan, not a media scan Update Figure 41, clarify that FFFFFFFFh is not supported for Get LBA Status command Bad Blocks generated by Write Uncorrectable command must be reported by the controller and will be specifically identifiable Number of Unrecoverable Logical Blocks, the estimate of the sum of the total number of potentially unrecoverable logical blocks in all of the namespaces identified in the LBA Status Log Namespace Elements in the LBA Status Information log page, is now in the header of the log page instead of broken out on a namespace-based element basis as it is more useful to know on a FRU basis. Clarify that clearing the event will cause the Aggregation Time and the Aggregation Threshold to restart. Various editorial changes including incorporating feedback from comments and WG meeting
02/14/2018	<ul style="list-style-type: none"> Support for Aggregation Time and Aggregation Threshold is made optional – i.e., controller doesn't have to support them Removed value of 0 for Number of LBA Range Descriptors (NLRD) – makes sense in LBA Status Descriptor but not LBA Range Descriptor 0 is a valid value for Number of LBA Status Log Namespace Elements Aggregation Time is now in millisecond increments. Aggregation Time and Aggregation Thresholds are no longer “minimums”. Removed aggregation attributes from Identify controller – just use set/get features.
02/20/2018	<ul style="list-style-type: none"> Accepted various editorial changes from Mike Allison Added text to section 8.NEW clarifying that if all bad blocks are not dealt, with a new AEN may not be generated on their behalf Added text to identify situation where the host should send GET LBA

	<p>Status commands for all ranges/every attached namespace and situation where host shouldn't send any commands because nothing's wrong (NLSLNE == 0 and NUMULB == 0 vs NLSLNE == 0 and NUMULB != 0)</p> <ul style="list-style-type: none"> • Completion Condition 03h applies to Action Type 11h • Made mention of component Failure use case in Foreground scan definition. • Added new Command Specific status code "Unsupported Action Type" • From Bill Martin: modifications to 5.New to create a name for the two types of scans for "bad" LBAs to address Mike's concerns about a definition of the scan types and to differentiate what Get LBA Status command does from what the Verify command does
02/28/2018	<ul style="list-style-type: none"> • Deleted Unsupported Action Type Error Code • Major rewrite of Get LBA Status bad block detection mechanisms/ differences between Action Types 10h and 11h • Integrated lots of feedback from Mike A. • Integrated feedback from Bill M. • Lots of editorial clean up
03/06/2018	<ul style="list-style-type: none"> • Replaced Virtual address info with VTM address info • Added support for "no bad blocks/do not send Get LBA Status commands" to the RTYPE field in the LBA Status Information Log Page • Clarified that the controller may (or may not) indicate that specific LBA Ranges which are returned are due to Write Uncorrectable command • Clarified that if the host re-writes Known LBAs, they are removed from any internal "pending defects" list the controller may be maintaining. Same goes for Discovered LBAs that are rewritten – they are removed from any internal Discovered or Known LBA lists the controller may be maintaining • Fixed lots of editorial errors • Modified wording of Completion Condition Code 02h in LBA Status Descriptor List to cover both 10h and 11h Action Type cases. This allowed for the removal of the "Applicable Action Type" column as Mike A pointed out, • Rewrote parts of section 8 to provide clarifications around the role aggregation conditions play, when/how to discover defaults and modify configurable conditions if desired, exactly when/how AENs are triggered and when Aggregation Time is reset.
03/18/2018	<ul style="list-style-type: none"> • Changed Known/Discoverable to Tracked/Untracked LBAs • Eliminated Aggregation Threshold as advertised/configurable setting. May still exist internally and may trigger event if exceeded. • Made Aggregation Time non-optional for controller to support; however, it now only applies to Tracked LBAs. Untracked LBAs (die/component failure,etc.) will be reported immediately • Lots of editorial cleanup • Introduced concept of Aggregation Time being sufficient for controller to guarantee sync-up/lazy write of Tracked LBA list between LBA Status Info Available events • Replaced implementation-specific language referring to placement into physical locations.
03/20/2018	<ul style="list-style-type: none"> • Updates from 03/20 subteam conference call • Replace occurrences of "Read command" with "read operation". Read Operation to be defined in John Geldman TP as covering the following 3 commands: read,compare, and verify (once ratified). •
03/26/2018	<ul style="list-style-type: none"> • Address Curtis Ballard and Bill Martin latest comments • Change Aggregation Time to Report Interval and changed the definition to convey that this is about synchronization of host re-writes and controller detection of those re-writes, not just intervals between AENs • Generation Counter – changed the definition and size • Changed the definition of the AEN itself to a not refer to a "new" log

	<p>page being “available”; rather that it is a notification that one or more potentially unrecoverable LBAs exist and in order to investigate, the host should read the LBA Status Information Log Page</p>
04/16/2018	<ul style="list-style-type: none"> • Moved RTYPE (Recommended Type) out of log page header and down into individual namespace elements instead as RTYPE may vary by NVM Set • Added Return Updated Tracked LBA List (RUTLL) bit to Get LBA Status command to force sync up of Tracked LBA List before completion of the command • Removed flow around doing all Get LBA Status cmds and all writes/recovery of bad LBAs before clearing LBA Status Information log page event – this is just one example of what a host may choose to do • Changed Report Interval to Sync Interval • Added section on sample host flows (informative) • Changed RTYPE to RATYPE (Recommended Action Type) since that is what it is referring to. • Re-wrote para on steps involved prior to using Get LBA Status capability in response to Fred's feedback that ordering was confusing/incorrect • Incorporated and addressed comments from Naren • Incorporated and addressed comments from Jim Hatfield
04/18/2018	<ul style="list-style-type: none"> • Incorporated feedback from 04/17/2018 sub-team con call • Removed Return Updated Tracked LBA List (RUTLL) bit; group decided that Get LBA Status command will always sync up Tracked LBA List before completion of the command – i.e., nothing will be reported that has already been successfully recovered • Made wording consistent in the Optional Asynchronous Events Supported (OAES) field • Resolved what to do about defining “Read Operation” by using the word “command” instead. The base spec does not list the commands that may return Unrecovered Read Error anywhere else in the spec so this TP should not need to either: <i>“If a read error occurs during the processing of a command, (e.g. End-to-end Guard Check Error, Unrecovered Read Error), the controller may either stop the DMA transfer into the memory or transfer the erroneous data to the memory.”</i>
05/06/2018	<ul style="list-style-type: none"> • Closed a lot of comments which have had lots of review w/no further feedback • Moved Get LBA Status command into the NVM Command Set specific Admin command range (0x8x) rather than the generic Admin command range • Changed Sync Interval back to Report Interval because this attribute will no longer be overloaded to mean both – synchronization is now guaranteed everytime the Get LBA Status command is completed. • Re-organized portions of section 5.New Get LBA Status per Jim Hatfield's feedback • Substantial rewrite of LBA Status Information Report Interval section • Eliminated Completion Condition 03h code from LBA Status Descriptor List data structure
05/08/2018	<ul style="list-style-type: none"> • Scrub doc after SubTeam conference call, save a lot of Phase 3 comments in separate doc
05/09/2018	<p>Judy & Bill:</p> <ul style="list-style-type: none"> • Added a second interval attribute and renamed the feature to LBA Status Attributes feature • Specified that a new AEN would only be generated if the generation counter is incremented.
05/29/2018	<p>Incorporated Phase 3 comments from Naren (WDC) Incorporated feedback from Edward (Silicon Motion) Modified filename to comply with new guidelines</p>
05/30/2018	<p>Incorporated editorial changes from Nick (Intel) & from Edward.</p>

06/10/2018	Incorporate feedback from 06/07/18 WG mtg., incorporate feedback from Bill Martin
06/12/2018	With offline sub-team input, closed on RAE language, "prior to processing" vs "prior to completion" language, math in LBA Status Descriptor Entry list. Added new noun "Potentially Unrecoverable LBA".
06/15/2018	Integrate modification to definition of "Potentially Unrecoverable LBA" suggested by Curtis. Accepted one previously-rejected modification from Bill.
06/18/2018	Integrated feedback from Mike Allison Numerous editorial changes from Judy Brock
06/27/2018	From Mike (Intel): change an instance of "Info" to "Information" From Edward (Silicon Motion): a) typo in the description of Figure 8.NEW.Figure_2. The Starting LBA should be '10' b) typo in Figure 5.14.1.NEW_Figure_2: Status Range Added text to clarify that Untracked LBA List will not be re-reported in LBA Status Information Log (will only be reported once)
07/30/2018	Integrated editorial feedback from 30 day member review
08/03/2018	Integrated into NVM Express 1_NEXTc refactoring 2018.08.03 – Redline – Draft.docx
08/15/2018	Accepted/Integrated changes from tech writer into this TP
08/19/2018	Removed yellow highlight in various places throughout the document. Integrated change on last page of document; added editorial change from Hugh Curley
09/06/2018	Moved LBA Status Information Notices event in the OAES field to bit 13 in order to align with Asynchronous Event Configuration – Command Dword 11
9/17/2018	Ratified

Description of Specification Changes

Modify a portion of Figure 109 (Identify – Identify Controller Data Structure) as shown below:

Figure 1: Identify – Identify Controller Data Structure

Bytes	O/M	Description
Controller Capabilities and Features		
01:00	M	PCI Vendor ID (VID): Contains the company vendor identifier that is assigned by the PCI SIG. This is the same value as reported in the ID register in section 2.1.1.
03:02	M	PCI Subsystem Vendor ID (SSVID): Contains the company vendor identifier that is assigned by the PCI SIG for the subsystem. This is the same value as reported in the SS register in section 2.1.17.
...		...
95:92	M	<p>Optional Asynchronous Events Supported (OAES): This field indicates the optional asynchronous events supported by the controller. A controller shall not send optional asynchronous events before they are enabled by host software.</p> <p>Bits 31:14 40 are reserved.</p> <p>Bit 13 is set to '1' if the controller supports the LBA Status Information Notices event. If cleared to '0', then the controller does not support the LBA Status Information Notices event.</p> <p>Editor's note: bit 12 is assigned to Predictable Latency Event Aggregate Log Change Notices event in TP 4003a.</p> <p>Bits 11:10 are reserved</p> <p>Bit 9 is set to '1' if the controller supports sending-the Firmware Activation Notices event. If cleared to '0' then the controller does not support the Firmware Activation Notices event.</p> <p>Bit 8 is set to '1' if the controller supports sending-the Namespace Attribute Notices event and the associated Changed Namespace List log page. If cleared to '0' then the controller does not support the Namespace Attribute Notices event nor the associated Changed Namespace List log page.</p> <p>Bits 7:0 are reserved.</p>
...		...
Admin Command Set Attributes & Optional Controller Capabilities		

Bytes	O/M	Description
257:256	M	<p>Optional Admin Command Support (OACS): This field indicates the optional Admin commands and features supported by the controller. Refer to section 5.</p> <p>Bits 15:109 are reserved.</p> <p>Bit 9 if set to '1', then the controller supports the Get LBA Status capability (refer to section 8.NEW). If cleared to '0', then the controller does not support the Get LBA Status capability.</p> <p>Bit 8 if set to '1' then the controller supports the Doorbell Buffer Config command. If cleared to '0' then the controller does not support the Doorbell Buffer Config command.</p> <p>Bit 7 if set to '1' then the controller supports the Virtualization Management command. If cleared to '0' then the controller does not support the Virtualization Management command.</p> <p>Bit 6 if set to '1' then the controller supports the NVMe-MI Send and NVMe-MI Receive commands. If cleared to '0' then the controller does not support the NVMe-MI Send and NVMe-MI Receive commands.</p> <p>Bit 5 if set to '1' then the controller supports Directives. If cleared to '0' then the controller does not support Directives. A controller that supports Directives shall support the Directive Send and Directive Receive commands. Refer to section 9.</p> <p>Bit 4 if set to '1' then the controller supports the Device Self-test command. If cleared to '0' then the controller does not support the Device Self-test command.</p> <p>Bit 3 if set to '1' then the controller supports the Namespace Management and Namespace Attachment commands. If cleared to '0' then the controller does not support the Namespace Management and Namespace Attachment commands.</p> <p>Bit 2 if set to '1' then the controller supports the Firmware Commit and Firmware Image Download commands. If cleared to '0' then the controller does not support the Firmware Commit and Firmware Image Download commands.</p> <p>Bit 1 if set to '1' then the controller supports the Format NVM command. If cleared to '0' then the controller does not support the Format NVM command.</p> <p>Bit 0 if set to '1' then the controller supports the Security Send and Security Receive commands. If cleared to '0' then the controller does not support the Security Send and Security Receive commands.</p>
...
1791:1024		Reserved

Modify a portion of Figure 42 (Opcodes for Admin Commands – NVM Command Set Specific) as shown below:

Figure 2: Opcodes for Admin Commands – NVM Command Set Specific

Opcode (07)	Opcode (06:02)	Opcode (01:00)	Opcode ²	O/M ¹	Namespace Identifier Used ³	Command
Generic Command	Function	Data Transfer ⁴				
1b	000 00b	00b	80h	O	Yes	Format NVM
1b	000 00b	01b	81h	O	NOTE 5	Security Send
1b	000 00b	10b	82h	O	NOTE 5	Security Receive
1b	000 01b	00b	84h	O	No	Sanitize
1b	000 01b	10b	86h	O	NOTE 6	Get LBA Status
NOTES: 1. O/M definition: O = Optional, M = Mandatory. 2. Opcodes not listed are reserved. 3. A subset of commands uses the Namespace Identifier field (CDW1.NSID). When not used, the field shall be cleared to 0h. 4. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional. 5. The use of the Namespace Identifier is Security Protocol specific. 6. This command does not support the use of the Namespace Identifier field (CDW1.NSID) set to FFFFFFFFh.						

Add section 5.New, as shown below:

5.New Get LBA Status command – NVM Command Set Specific

The Get LBA Status command requests information about Potentially Unrecoverable LBAs (refer to section 8.NEW). If the Get LBA Status command completes successfully, then the LBA Status Descriptor List, defined in Figure 5.New_5, is returned in the data buffer for that command.

The Get LBA Status command uses the Data Pointer, Command Dword 10, Command Dword 11, Command Dword 12, and Command Dword 13 fields. All other command specific fields are reserved.

The Maximum Number of Dwords (MNDW) field contains the maximum number of Dwords to return. Upon successful command completion, the actual amount of data returned by the controller is indicated by the Number of LBA Status Descriptors (NLSD) field in the LBA Status Descriptor List.

A controller identifies Potentially Unrecoverable LBAs using the following two report types:

- Tracked LBAs:** a list of Potentially Unrecoverable LBAs associated with physical storage. These may be discovered through a background scan where the controller examines the media in the background or discovered through other means. The Tracked LBA list is able to be returned without significant delay; or
- Untracked LBAs:** a list of Potentially Unrecoverable LBAs generated by a scan originated by a Get LBA Status command with the ATYPE field set to 10h. The controller scans internal data structures related to LBAs to determine which LBAs are Potentially Unrecoverable LBAs. The controller may use this scan to determine which LBAs in which namespaces are affected by a component (e.g., die or channel) failure. Significant delays may be incurred during the processing of a Get LBA Status

command with the ATYPE field set to 10h. After discovery of Untracked LBAs, they may or may not be added to the list of Tracked LBAs.

In response to a Get LBA Status command, the controller shall return LBA Status Descriptors that describe LBAs written by a Write Uncorrectable command in addition to any other LBAs that may return an Unrecovered Read Error status discovered through other mechanisms. The list of Tracked LBAs and the list of Untracked LBAs may include LBA Status Descriptor Entries that describe LBAs written by a Write Uncorrectable command. If an LBA Status Descriptor Entry describes LBAs written by a Write Uncorrectable command, then bit 1 in the Status field of that entry should be set to '1'.

Figure 5.New_1: Get LBA Status – Data Pointer

Bit	Description
127:00	Data Pointer (DPTR): This field specifies the start of the data buffer. Refer to Figure 11 for the definition of this field.

Figure 5.New_2: Get LBA Status – Command Dword 10 and Command Dword 11

Bit	Description
63:00	Starting LBA (SLBA): This field indicates the 64-bit address of the first logical block addressed by this command. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:32.

Figure 5.New_3: Get LBA Status – Command Dword 12

Bit	Description
31:00	Maximum Number of Dwords (MNDW): This field specifies the maximum number of Dwords to return. This is a 0's based value.

Figure 5.New_4: Get LBA Status – Command Dword 13

Bit	Description	
31:24	Action Type (ATYPE): This field specifies the mechanism the controller uses in determining the LBA Status Descriptors to return as defined in Figure 5.New_6.	
	Value	Description
	10h	Perform a scan and return Untracked LBAs and Tracked LBAs in the specified range
	11h	Return Tracked LBAs in the specified range
	All others	Reserved
23:16	Reserved	
15:00	Range Length (RL): This field specifies the length of the range of contiguous LBAs, beginning at Starting LBA (SLBA), that the action specified in the Action Type (ATYPE) field shall be performed on. A value of 0h in this field specifies the length of a range beginning at Starting LBA and ending at Namespace Size (NSZE) minus 1 (refer to Figure 114).	

If the value in the Action Type (ATYPE) field is 10h, then:

- the controller shall return Untracked LBAs and Tracked LBAs in the range specified in the Get LBA Status command for the namespace specified in the Namespace Identifier (CDW1.NSID);
- the controller shall remove all LBAs in the range specified in the Get LBA Status command, which prior to processing the Get LBA Status command were successfully re-written, from relevant internal data structures (e.g., internal Tracked LBA list);
- the controller shall ensure that any such successfully re-written logical blocks are not reported in any LBA Status Descriptor entries returned by the Get LBA Status command unless, after having been

removed from relevant internal data structures and prior to processing the Get LBA Status command, were newly detected as being Potentially Unrecoverable LBAs; and

- d) the list of Untracked LBAs returned by the Get LBA Status command may be discarded by the controller or added to the Tracked LBA list once the command has completed.

If the value in the Action Type field (ATYPE) is 11h, then the controller shall:

- return Tracked LBAs in the range specified in the Get LBA Status command for the namespace specified in the Namespace Identifier (CDW1.NSID) field;
- remove all LBAs in the range specified in the Get LBA Status command, which prior to processing the Get LBA Status command were successfully re-written, from relevant internal data structures (e.g., internal Tracked LBA list);
- ensure that any such successfully re-written logical blocks are not reported in any LBA Status Descriptor entries returned by the Get LBA Status command unless, after having been removed from relevant internal data structures and prior to processing the Get LBA Status command, were newly detected as being Potentially Unrecoverable LBAs; and
- not perform a foreground scan to generate and return Untracked LBAs.

Figure 5.New_5: LBA Status Descriptor List

Bytes	Description										
03:00	Number of LBA Status Descriptors (NLSD): This field indicates the number of LBA Status Descriptor entries returned by the controller in this data structure. An LBA Status Descriptor List which indicates that no LBA Status Descriptor entries have been returned (i.e., NLSD is cleared to '0') is a valid LBA Status Descriptor List.										
04	<p>Completion Condition (CMPC): If the command completes successfully, then this field indicates the condition that caused completion of the GET LBA STATUS command. If there are no more LBA Status Descriptor Entries to transfer in the specified range, the controller shall return Completion Condition 2h.</p> <table> <tr> <th>Code</th><th>Definition</th></tr> <tr> <td>0h</td><td>No indication of the completion condition.</td></tr> <tr> <td>1h</td><td>The command completed due to transferring the amount of data specified in the MNDW field. There may be more LBA Status Descriptor Entries to transfer in the specified range.</td></tr> <tr> <td>2h</td><td>The command completed due to having performed the action specified in the Action Type field over the number of logical blocks specified in the Range Length field. There are no more LBA Status Descriptor Entries to transfer in the specified range.</td></tr> <tr> <td>All others</td><td>Reserved</td></tr> </table>	Code	Definition	0h	No indication of the completion condition.	1h	The command completed due to transferring the amount of data specified in the MNDW field. There may be more LBA Status Descriptor Entries to transfer in the specified range.	2h	The command completed due to having performed the action specified in the Action Type field over the number of logical blocks specified in the Range Length field. There are no more LBA Status Descriptor Entries to transfer in the specified range.	All others	Reserved
Code	Definition										
0h	No indication of the completion condition.										
1h	The command completed due to transferring the amount of data specified in the MNDW field. There may be more LBA Status Descriptor Entries to transfer in the specified range.										
2h	The command completed due to having performed the action specified in the Action Type field over the number of logical blocks specified in the Range Length field. There are no more LBA Status Descriptor Entries to transfer in the specified range.										
All others	Reserved										
07:05	Reserved										
23:08	LBA Status Descriptor Entry 0: This field contains the first LBA Status Descriptor Entry in the list, if present.										
39:24	LBA Status Descriptor Entry 1: This field contains the second LBA Status Descriptor Entry in the list, if present.										
...	...										
(N*16+23): (N*16+8)	LBA Status Descriptor Entry N: This field contains the N+1 LBA Status Descriptor Entry in the list, if present.										

Figure 5.New_6: LBA Status Descriptor Entry

Bytes	Description								
07:00	Descriptor Starting LBA (DSLBA): This field specifies the 64-bit address of the first logical block of the LBA range for which this LBA Status Descriptor reports LBA status.								
11:08	Number of Logical Blocks (NLB): This field contains the number of contiguous logical blocks reported in this LBA Status Descriptor. The controller should return the largest aggregated possible value in this field. This is a 0's based value.								
12	Reserved								
13	Status: This field contains additional status about this LBA range. <table> <tr> <th>Bits</th><th>Definition</th></tr> <tr> <td>7:2</td><td>Reserved</td></tr> <tr> <td>1</td><td> If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command. </td></tr> <tr> <td>0</td><td> If set to '1', a read, verify, or compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a read, verify, or compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. </td></tr> </table>	Bits	Definition	7:2	Reserved	1	If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.	0	If set to '1', a read, verify, or compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a read, verify, or compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.
Bits	Definition								
7:2	Reserved								
1	If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.								
0	If set to '1', a read, verify, or compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a read, verify, or compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.								
15:14	Reserved								

The Descriptor Starting LBA (DSLBA) field in the first LBA Status Descriptor entry returned in the LBA Status Descriptor List shall contain the lowest numbered LBA that is greater than or equal to the value specified in the Starting LBA field in the Get LBA Status command.

For subsequent LBA Status Descriptor entries, the contents of the Descriptor Starting LBA field shall contain the value of the lowest numbered LBA meeting the requirements for the specified Action Type value that is greater than the sum of the values in:

- the Descriptor Starting LBA field in the previous LBA Status Descriptor entry; and
- the Number of Logical Blocks field in the previous LBA Status Descriptor entry.

5.New.1 Command Completion

When the command is completed, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.

Modify a portion of Figure 49 (Asynchronous Event Information - Notice) as shown below:

Figure 49: Asynchronous Event Information - Notice

Value	Description
0h	<p>Namespace Attribute Changed: The Identify Namespace data structure for one or more namespaces, as well as the Namespace List returned when the Identify command is issued with the CNS field set to 02h, have changed. Host software may use this event as an indication that it should read the Identify Namespace data structures for each namespace to determine what has changed.</p> <p>Alternatively, host software may request the Changed Namespace List (Log Identifier 04h) to determine which namespaces in this controller have changed Identify Namespace information since the last time the log page was read.</p> <p>A controller shall not send this event when Namespace Utilization has changed, as this is a frequent event that does not require action by the host. A controller shall only send this event for changes to the Format Progress Indicator field when bits 6:0 of that field transition from a non-zero value to zero, or from a zero value to a non-zero value.</p>
1h	<p>Firmware Activation Starting: The controller is starting a firmware activation process during which command processing is paused. Host software may use CSTS.PP to determine when command processing has resumed. To clear this event, host software reads the Firmware Slot Information log page.</p>
2h	<p>Telemetry Log Changed: The controller has saved the controller internal state in the Telemetry Controller-Initiated log page and set the Telemetry Controller-Initiated Data Available field to 1h in that log page. To clear this event, the host issues a Get Log Page command with Retain Asynchronous Event cleared to '0' for the Telemetry Controller-Initiated Log.</p>
5h	<p>LBA Status Information Alert: The criteria for generating an LBA Status Information Alert Notice event have been met (refer to section 8.NEW). Information about Potentially Unrecoverable LBAs is available in the LBA Status Information log page (refer to section 5.14.1.NEW). To clear this event, the host issues a Get Log Page command with Retain Asynchronous Event cleared to '0' for the LBA Status Information log.</p>
36h – FFh	Reserved

Modify a portion of Figure 90 (Get Log Page – Log Page Identifiers) as shown below:

Figure 3: Get Log Page – Log Page Identifiers

Log Identifier	O/M	Scope	Description	Reference Section
00h	Reserved			
01h	M	Controller	Error Information	5.14.1.1
02h	M	NVM subsystem ¹	SMART / Health Information	5.14.1.2
	O	Namespace ²		
03h	M	NVM subsystem	Firmware Slot Information	5.14.1.3
04h	O	Controller	Changed Namespace List	5.14.1.4
05h	O	Controller	Commands Supported and Effects	5.14.1.5
06h	O	NVM subsystem	Device Self-test	5.14.1.6
07h	O	Controller	Telemetry Host-Initiated	5.14.1.7
08h	O	Controller	Telemetry Controller-Initiated	5.14.1.8
09h	O	NVM subsystem	Endurance Group Information	5.14.1.9
0Eh	O	Controller	LBA Status Information	5.14.1.NEW
0A0Fh – 6Fh	Reserved			
70h	Discovery (refer to the NVMe over Fabrics specification)			
71h – 7Fh	Reserved for NVMe over Fabrics			

Add section 5.14.1.NEW, as shown below:

5.14.1.NEW LBA Status Information (Log Identifier 0Eh)

This log page is used to provide information about subsequent actions the host may take to discover which logical blocks, in namespaces that are attached to the controller, may no longer be recoverable when read. It contains zero or more LBA Status Log Namespace Elements (refer to Figure 5.14.1.NEW_Figure_2). If the controller is unaware of any potentially unrecoverable logical blocks in a given namespace attached to the controller, then this log page will not return an LBA Status Log Namespace Element for that namespace. This log page shall not return any LBA Status Log Namespace Elements for namespaces which are not attached to the controller.

Each LBA Status Log Namespace Element contains zero or more LBA Range Descriptors (refer to Figure 5.14.1.NEW_Figure_3). Each LBA Range Descriptor describes a range of LBAs that have been detected as being potentially unrecoverable and should be examined by the host using the mechanism specified in the Recommended Action Type field (refer to Figure 5.14.1.NEW_Figure_1) in that LBA Status Log Namespace Element in a subsequent Get LBA Status command.

The host may identify logical blocks that may no longer be recoverable through the subsequent issuing of one or more Get LBA Status commands. Once identified, the host may then recover the user data from an alternative source and write that data to the original logical block address in the namespace. If the user data is written successfully, subsequent reads should not cause unrecoverable read errors (e.g., as a result of the write changing the physical location of the user data).

Upon receiving an LBA Status Information Alert asynchronous event, the host should send one or more Get Log Page commands for Log Identifier 0Eh with the Retain Asynchronous Event bit set to '1' until it has read the entire log page. In order to clear the event, the host sends another Get Log Page command for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0'. The host decides when to send Get LBA Status commands and when to recover the LBAs identified by the Get LBA Status commands, relative to when it clears the LBA Status Information Alert asynchronous event. Section 8.NEW.1 describes example host implementations. Clearing the event causes the LBA Status Information Report Interval to be restarted and allows the contents of the log page to be updated.

Figure 5.14.1.NEW_Figure_1: LBA Status Information Log

Bytes	Description
03:00	LBA Status Log Page Length (LSLPLEN): This field indicates the length in bytes of the LBA Status Information log page.
07:04	Number of LBA Status Log Namespace Elements (NLSLNE): This field indicates the number of LBA Status Log Namespace Elements (refer to Figure 5.14.1.NEW_Figure_2) contained in this log. If this field is cleared to 0h and the Estimate of Unrecoverable Logical Blocks (ESTULB) field contains a non-zero value, the host should send Get LBA Status commands for the entire LBA range of each namespace attached to the controller. If both this field and the Estimate of Unrecoverable Logical Blocks (ESTULB) are cleared to 0h, the host should not send any Get LBA Status commands for any LBA ranges on any namespaces attached to the controller as there are no known potentially unrecoverable logical blocks in any namespace attached to the controller.
11:08	Estimate of Unrecoverable Logical Blocks (ESTULB): This field is an estimate of the sum of the total number of potentially unrecoverable logical blocks in all of the namespaces identified in the LBA Status Log Namespace Elements in this log page. A value of 0h in this field is valid. A value of FFFFFFFFh indicates no information regarding an estimate of the total number of potentially unrecoverable logical blocks is available.
13:12	Reserved

Bytes	Description
15:14	LBA Status Generation Counter (LSGC): Contains a value that is incremented each time the LBA Status Log contains one or more LBA Range Descriptors which specify any potentially unrecoverable logical blocks which were not included in any LBA Range Descriptors the last time the host read the LBA Status Information log. This field is persistent across power on and rolls over to zero when the maximum count is reached.
n:16	LBA Status Log Namespace Element List: This field contains the list of LBA Status Log Namespace Elements that are present in the log page, if any. LBA Status Log Namespace Elements are of variable length (see Figure 5.14.1.NEW_Figure_2).

Figure 5.14.1.NEW_Figure_2: LBA Status Log Namespace Element

Bytes	Description
03:00	Namespace Element Identifier (NEID): This field indicates the Namespace Identifier (NSID) of the namespace that this LBA Status Log Namespace Element applies to.
07:04	Number of LBA Range Descriptors (NLRD): This field indicates the number of LBA Range Descriptors (refer to Figure 5.14.1.NEW_Figure_3) returned by the controller in this LBA Status Log Namespace Element. A value of FFFFFFFFh indicates that: a) no LBA Range Descriptors are present; b) there is no information available regarding the location of known potentially unrecoverable blocks in the namespace; and c) the host should examine all LBAs in the namespace.
08	Recommended Action Type (RATYPE): This field indicates the value the host should set the Action Type (ATYPE) field to in Get LBA Status commands associated with LBA Range Descriptors contained in this LBA Status Log Namespace Element.
15:09	Reserved
31:16	LBA Range Descriptor 0: This field contains the first LBA Range Descriptor in this LBA Status Log Namespace Element, if present.
47:32	LBA Range Descriptor 1: This field contains the second LBA Range Descriptor in this LBA Status Log Namespace Element, if present.
...	...
(N*16+31): (N*16+16)	LBA Range Descriptor N: This field contains the N+1 LBA Range Descriptor in this LBA Status Log Namespace Element, if present.

Figure 5.14.1.NEW.Figure_3: LBA Range Descriptor

Bytes	Description
07:00	Range Starting LBA (RSLBA): This field specifies the 64-bit address of the first logical block of this LBA Range.
11:08	Range Number of Logical Blocks (RNLB): This field contains the number of logical blocks in this LBA Range. The controller should return the largest possible value in this field. This is a 0's based value.
15:12	Reserved

For a given LBA Status Log Namespace Element, if the value in the Recommended Action Type (RATYPE) field is 10h, then the controller shall not report the same LBA Status Log Namespace Element once the host issues a Get Log Page command for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0' unless an additional component failure has occurred that may have created additional Untracked LBAs.

Modify a portion of Figure 143 (Asynchronous Event Configuration – Command Dword 11) as shown below:

Figure 4: Asynchronous Event Configuration – Command Dword 11

Bit	Description
31:14	Reserved
13	LBA Status Information Notices: This bit determines whether an asynchronous event notification is sent to the host for an LBA Status Information Alert event (refer to Figure 49). If this bit is set to '1', then the LBA Status Information Alert event is sent to the host when this condition occurs. If this bit is cleared to '0', then the controller shall not send the LBA Status Information Alert event to the host.
10	Telemetry Log Notices: This bit determines whether an asynchronous event notification is sent to the host when the Telemetry Controller-Initiated Data Available field transitions from '0' to '1' in the Telemetry Controller-Initiated log page. If this bit is set to '1', then the Telemetry Log Changed event is sent to the host when this condition occurs. If this bit is cleared to '0', then the controller shall not send the Telemetry Log Changed event to the host.
09	Firmware Activation Notices: This bit determines whether an asynchronous event notification is sent to the host for a Firmware Activation Starting event (refer to Figure 49). If this bit is set to '1', then the Firmware Activation Starting event is sent to the host when this condition occurs. If this bit is cleared to '0', then the controller shall not send the Firmware Activation Starting event to the host.
08	Namespace Attribute Notices: This bit determines whether an asynchronous event notification is sent to the host for a Namespace Attribute change (refer to Figure 49). If this bit is set to '1', then the Namespace Attribute Changed event is sent to the host when this condition occurs. If this bit is cleared to '0', then the controller shall not send the Namespace Attribute Changed event to the host.
07:00	SMART / Health Critical Warnings: This field determines whether an asynchronous event notification is sent to the host for the corresponding Critical Warning specified in the SMART / Health Information Log (refer to Figure 96). If a bit is set to '1', then an asynchronous event notification is sent when the corresponding critical warning bit is set to '1' in the SMART / Health Information Log. If a bit is cleared to '0', then an asynchronous event notification is not sent when the corresponding critical warning bit is set to '1' in the SMART / Health Information Log.

Modify a portion of Figure 84 (Get Features – Feature Identifiers) as shown below:

Figure 5: Get Features – Feature Identifiers

Description	Section Defining Format of Attributes Returned
Arbitration	Section 5.21.1.1
...	...
Read Recovery Level Config	Section 5.21.1.18
LBA Status Information Report Interval	Section 5.21.1.NEW

Modify a portion of Figure 128 (Set Features – Feature Identifiers) as shown below:

Figure 6: Set Features – Feature Identifiers

Feature Identifier	O/M ⁶	Persistent Across Power Cycle and Reset ²	Uses Memory Buffer for Attributes	Description
00h				Reserved
01h	M	No	No	Arbitration
...
12h	O	Yes	No	Read Recovery Level Config

Feature Identifier	O/M ⁶	Persistent Across Power Cycle and Reset ²	Uses Memory Buffer for Attributes	Description
15h	O	No	No	LBA Status Information Attributes
43h 16h – 77h				Reserved
78h – 7Fh		Refer to the NVMe Management Interface Specification for definition.		
80h – BFh				Command Set Specific (Reserved)

Add new section 5.21.1.NEW, as shown below:

5.21.1.NEW LBA Status Information Attributes (Feature Identifier 15h)

The LBA Status Information Poll Interval (LSIPI) (refer to Figure 5.21.NEW_Figure_1) is the minimum interval that the host should wait between subsequent reads of the LBA Status Information log page with the Retain Asynchronous Event bit cleared to '0'. The LBA Status Information Poll Interval (LSIPI) is not changeable by the host.

The LBA Status Information Report Interval (LSIRI) (refer to Figure 5.21.NEW_Figure_1) is the minimum amount of time that a controller shall delay before sending an LBA Status Information Alert asynchronous event, if LBA Status Information Notices are enabled. The default value of the LSIRI is equal to LSIPI.

The host may read the LBA Status Information log page as part of LBA Status Information Alert asynchronous event processing or it may read it in a polled fashion without enabling LBA Status Information Notices.

The controller reports the value of the LBA Status Information Attributes in Dword 0 of the completion queue entry when the host issues either a Set Features or Get Features command for this feature. The host configures the LBA Status Information Report Interval by issuing a Set Features command for this feature and specifying the desired value of the LBA Status Information Report Interval in Command Dword 11 (refer to Figure 5.21.NEW_Figure_1).

The host should not specify a value for the LBA Status Information Report Interval (LSIRI) which is less than the LBA Status Information Poll Interval (LSIPI) value reported by the controller. If the host specifies a value the controller does not support, the controller shall return the closest value it supports in Dword 0 of the completion queue entry for the Set Features command. The accuracy of the interval measurement on the part of the controller is implementation specific.

The controller shall not send an LBA Status Information Alert asynchronous event unless:

- a) there are Tracked LBAs and:
 - a. the LBA Status Information Report Interval condition has been exceeded and the LBA Status Generation Counter has been incremented since the last LBA Status Information Alert asynchronous event occurred; or
 - b. an implementation specific aggregate threshold, if any exists, of Tracked LBAs has been exceeded;
- or
- b) a component (e.g., die or channel) failure has occurred that may result in the controller aborting commands with Unrecovered Read Error status.

When the host issues a Get Log Page command for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0', the LBA Status Information Alert asynchronous event is cleared, if one was outstanding, and the LBA Status Information Report Interval is restarted by the controller.

LBAs added to the Tracked LBA List or component failures that generate potential LBAs for an Untracked LBA list may be coalesced into a single LBA Status Information Alert asynchronous event notification.

Figure 5.21.NEW_Figure_1: LBA Status Information Attributes – Command Dword 11

Bit	Description
31:16	LBA Status Information Poll Interval (LSIPI): The minimum amount of time in 100 millisecond increments that the host should wait between subsequent reads of the LBA Status Information log page with the Retain Asynchronous Event bit cleared to '0'.
15:00	LBA Status Information Report Interval (LSIRI): If LBA Status Information Notices are enabled, the value in this field is the minimum amount of time in 100 millisecond increments that a controller shall delay before sending an LBA Status Information Alert asynchronous event.

Add new section 8.NEW, as shown below:

8.NEW Get LBA Status

Potentially Unrecoverable LBAs are LBAs that, when read, may result in the command that caused the media to be read being aborted with Unrecovered Read Error status. The Get LBA Status capability provides the host with the ability to identify Potentially Unrecoverable LBAs. The logical block data can then be recovered from another location and re-written.

To support the Get LBA Status capability, the NVM subsystem shall:

- indicate support for the Get LBA Status capability in the Optional Admin Command Support (OACS) field in the Identify Controller data structure;
- indicate support for LBA Status Information Notices in the Optional Asynchronous Events Supported field in the Identify Controller data structure;
- support the LBA Status Information log page;
- indicate support for the Log Page Offset and extended Number of Dwords (i.e., 32 bits rather than 12 bits) in the Log Page Attributes field of the Identify Controller data structure;
- support the LBA Status Attributes Feature;
- support the Get LBA Status command; and
- support the LBA Status Information Alert Notices event.

Prior to using the Get LBA Status capability:

- The host should use the Get Features and Set Features commands with the LBA Status Attributes Feature (refer to section **5.21.1.NEW**) to retrieve and optionally configure the LBA Status Information Report Interval attribute; and
- If the host wishes to receive LBA Status Information Alert asynchronous events, the host should enable LBA Status Information Notices (refer to Figure 148).

If LBA Status Information Notices are enabled, the controller shall send an LBA Status Information Alert asynchronous event if:

- a) there are Tracked LBAs and:
 - a. the LBA Status Information Report Interval condition has been exceeded; or
 - b. an implementation specific aggregate threshold, if any exists, of Tracked LBAs has been exceeded;
- or
- b) a component (e.g., die or channel) failure has occurred that may result in the controller aborting commands with Unrecovered Read Error status.

Upon receiving an LBA Status Information Alert asynchronous event, the host should send one or more Get Log Page commands for Log Identifier 0Eh with the Retain Asynchronous Event bit set to '1' in order to read the entire LBA Status Information log page (refer to section 5.14.1.NEW).

Once the host has started reading the LBA Status Information log page with the Retain Asynchronous Event bit set to '1', the controller shall not modify the contents of that log page until the host re-reads the LBA Status Information log page with the Retain Asynchronous Event bit cleared to '0'.

The LBA Status Information log page returns zero or more sets of per-namespace LBA Range Descriptors. Each LBA Range Descriptor specifies a range of LBAs that should be examined by the host in a subsequent Get LBA Status command (refer to section 5.New).

The Get LBA Status command requests information about Potentially Unrecoverable LBAs in a specified range.

The LBA Status Information Report Interval is restarted by the controller when the host issues a Get Log Page for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0'. Reading the Get Log Page for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0' will also cause an outstanding LBA Status Information asynchronous event to be cleared, if there is one outstanding on the controller.

When the host re-reads the header of the LBA Status Information log page with the Retain Asynchronous Event bit cleared to '0', the host should ensure that the LBA Status Generation Counter matches the original value read. If it does not match, there is newer LBA Status Log Page data available than that which was returned to the host the last time the host read the LBA Status Information log. In this case, the host need not wait for the LBA Status Information Poll Interval (LSIPI) to pass before re-reading the LBA Status Information log page.

The host decides when to send Get LBA Status commands and when to recover the LBAs identified by the Get LBA Status commands, relative to when it reads the Get Log Page for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0'. Section 8.NEW.1 describes some example host implementations.

The Get LBA Status command may return zero or more LBA Status Descriptors (refer to Figure 5.New_5) for each LBA Range Descriptor (refer to Figure 5.14.1.10.Figure_New_3) returned by the LBA Status Information log page.

Figure 8.NEW.Figure_1: Example LBA Range Identifiers returned by LBA Status Information Log Page

Bytes	Description	Value		
03:00	Namespace Element Identifier	1		
07:04	Number of LBA Range Descriptors	2		
15:08	Reserved			
31:16	LBA Range Descriptor 0	Description	Value	
		Range Starting LBA	10	
		Range Number of Logical Blocks	1000	
47:32	LBA Range Descriptor 1	Description	Value	
		Range Starting LBA	15000	
		Range Number of Logical Blocks	15010	

Figure 8.NEW.Figure_2: Example LBA Status Descriptors for Get LBA Status Command issued for LBA Range Descriptor 0 in Figure 8.NEW.Figure_1 (Starting LBA set to 10, Range Length set to 1000)

Bytes	Description	Value		
03:00	Number of LBA Status Descriptors	3		
04	Completion Condition:	2		
07:05	Reserved			
23:08	LBA Status Descriptor Entry 0	Description	Value	
		Descriptor Starting LBA	10	
		Number of Logical Blocks	30	
39:24	LBA Status Descriptor Entry 1	Description	Value	
		Descriptor Starting LBA	550	
		Number of Logical Blocks	75	
55:40	LBA Status Descriptor Entry 2	Description	Value	
		Descriptor Starting LBA	1000	
		Number of Logical Blocks	10	

Figure 8.NEW.Figure_3: Example LBA Status Descriptors for Get LBA Status Command issued for LBA Range Descriptor 1 in Figure 8.NEW.Figure_1 (Starting LBA set to 15000, Range Length set to 15010)

Bytes	Description	Value		
03:00	Number of LBA Status Descriptors	1		
04	Completion Condition	2		
07:05	Reserved			
23:08	LBA Status Descriptor Entry 0	Description	Value	
		Descriptor Starting LBA	15000	
		Number of Logical Blocks	15010	

8.NEW.1 Sample Get LBA Status Host Software Implementations (Informative)

8.NEW.1.1 Example Flow #1

- 1) Read all the LBA Status Information log page data with RAE bit set to '1';
- 2) Complete all necessary Get LBA Status commands;
- 3) Complete all necessary recovery of the affected user data by rewriting that data; and
- 4) Read the LBA Status Information log page data with RAE bit cleared to '0'.

8.NEW.1.2 Example Flow #2

- 1) Read all the LBA Status Information log page data with RAE bit set to '1';
- 2) Read the LBA Status Information log page data with RAE bit cleared to '0';
- 3) Issue some host-determined subset of the Get LBA Status commands indicated by the log page data;
- 4) Complete the recovery of the affected user data returned by the Get LBA Status commands issued so far;
- 5) Re-issue the Get LBA Status commands over the ranges associated with the re-written (i.e., recovered) user data;
- 6) Confirm that the re-written LBAs are no longer in the Tracked LBA List (if any are still there, they are there because they have been detected as newly bad again);
- 7) Add any new LBA ranges returned in the Get LBA Status commands to the list of LBAs still outstanding the host needs to recover; and
- 8) If the host has not processed all LBA ranges returned by:
 - o the LBA Status Information log page in step 1); and
 - o the Get LBA Status command(s) in step 7),then go back to step 3).