LEGAL NOTICE:

LEGAL DISCLAIMER:

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

NVM Express™ Technical Errata

Errata ID	**003**
Revision Date	**2020-05-28**
Affected Spec Ver.	**NVM Express 1.4**
Corrected Spec Ver.	**NVM Express 1.4+**

Errata Author(s)

Company	**Authors**
Seagate	Jim Hatfield
Broadcom	Brad Besmer
Dell EMC	Kevin Marks, David Black, Austin Bolen
Intel	Mike Allison, Nick Adams
NetApp	Fred Knight
Samsung	Judy Brock
Western Digital	Christoph Hellwig, Yoni Shternhell

Errata Overview

Misc corrections and clarifications to NVMe 1.4

Revision History

Revision Date	Change Description
2019-12-19	Initial draft
2020-01-02	1. Included the remainder of the changes and discussion points in the PDF initially used to gather inputs. 2. And included new change requests from the reflector as of today.
2020-01-23	Changed marking of inserted text from red to blue underscore for easier reading. As the result of discussion in the 1/23/20 workgroup meeting: 1. Resolved all instances of 'this controller', resulting in some proposed changes being removed 2. Resolved changes to figure 147 re: the Namespace Attribute Changed notice. 3. Resolved that the Error Count in figure 193 wraps to '1' and does not saturate 4. Resolved that byte 00, bit 3 intends to mean 'all of the media' 5. Removed changes that are already in ECN 001 6. Removed proposed changes to figure 200, as the group determined that the proposed change would require a new TP 7. Resolved that the first sentence of 5.14.1.10 needed to change, and proposed text was suggested in the meeting 8. Removed proposed change to sections 4.3, as the current text was deemed to be satisfactory. 9. Removed proposed change figures 115, 118 as the current text was deemed to be satisfactory. 10. Removed comments for items that have been addressed and have not been contested Note: I still need to integrate the change requests on the last page of this document.
2020-01-24	1. Change all SFSC references to SPC-4: sections 1.9, 2.25, 2.26 2. Section 5.13 Get Features command 3. Section 5.14.1.2 SMART / Health Information 4. Figure 134: NVM Set Aware Admin Commands 5. Figure 245: Identify – Identify Namespace Data Structure (NVMCAP) 6. PI-related changes in 6.17 and 8.3.1.5
2020-01-30	1. Resolved comments in sections: 5.14.1.2, 2. Global resolution about 'saved' vs. 'saveable': a. 'saved' is the value, 'saveable' is the characteristic b. Carefully considered each instance c. Modified additional sections as required: 15.13.1, 15.13.2, 5.21.1, 5.21.1.14, 5.21.1.22, 5.21.1.23, 7.8, 3. Updated proposed changes to Namespace Attribute Changed in Figure 142, per the email thread about that topic. 4. Resolved comments in Figures 193, 382
2020-02-05	1. Added change to Figure 129 2. Added change to section 8.3.1.5 (from Christoph) 3. Filled in the 'summary of changes' table. Note: yellow items are not fully resolved yet. 4. Added change to figure 159 5. Further updates to figure 147, based on reflector discussions

Revision Date	Change Description
2020-02-06	As the result of discussion in the WG meeting: 1. Determined that some fixes were already handled in ECN001, so they have been removed from this ECN: a. 6.17 2. Added some comments from Mike Allison for sections: a. Figure NEW, figure 193, figure 210, 5.21.1.14, 5.21.1.22 3. Resolutions were suggested for 5.19, figure 330, figure 262 4. Paul Suhler will propose the change for 8.12.1 5. New item from Fred Knight: figure 329 6. New item from David Black/Fred Knight: figure 494 7. Updated the list of authors 8. WG decided to not accept any additional changes into this ECN. New corrections/clarifications will go into a future ECN.
2020-02-10	1. Moved proposed changes to ECN 004 a. 5.13 Get Features command b. 5.14.1.10 Predictable Latency Event log page c. 5.14.1.11 Predictable Latency Event Aggregate log page d. 5.14.1.12 Asymmetric Namespace Access log page e. 5.22.1 Virtualization Management cmd sent to secondary ctlr
2020-03-05	1. Moved changes to 8.12.1 Namespace granularity to ECN 004 2. Resolved comments about Figure 330: Sanitize – Command Dword 10 3.
2020-03-11	1. Section 5.24: Resolved how to handle Sanitize (exit failure mode) when a) There is no sanitize operation in process; and b) When a sanitize operation is in process (but has not failed)
2020-03-12	1. Added section 5.24 to the list of incompatible changes
2020-03-19	2. Addressed member review comments from Fred Knight and Mike Allison
2020-04-06	1. Finished addressing member review comments from Fred Knight
2020-05-08	1. Changed tables of incompatible changes and summary of changes to bullet lists
2020-05-27	1. Integrated into NVMe Base Specification

Incompatible Changes

- In the Error Count field of the Error Information Log Entry,
  - NVMe 1.4 did not specify what happens if the value of the field is FFFFFFFFh and increments again.
  - This ECN specifies that it shall wrap to 00000001h.
- In the Establish Context and Read Log Data field in Command Dword 10 of the Get Log Page command:
  - NVMe 1.4 specified that no log data shall be returned if the number of dwords to be returned is set to zero
  - This ECN deletes that requirement
- In the Sanitize command:
  - NVMe 1.4 did not specify how to process Sanitize Sanitize Action set to 001b (i.e., Exit Failure Mode) when Sanitize is not in process and has not failed.
  - This ECN requires that the command shall complete with a status of Successful Completion and perform no other action

Summary of changes:

- Improved consistency: Misc corrections of capitalization, spelling, punctuation, whitespace, cross references,
- References: Remove reference for SFSC to SPC-4; modify TCG SIIS reference title
- Clarified the meaning of ‘this controller’ where it was ambiguous
- Changed ‘fail’ to ‘abort’
- Changed ‘indicated’ to ‘specified’ as appropriate
- Changed ‘saved’ to ‘saveable’ as appropriate
- Changed ‘degrees Kelvin’ to ‘Kelvins’
- Add Flush and Format NVM as commands that may terminate with status of Attempted Write to Read

Only Range

- Indicate that Get Features command is also NVM Set aware, and that two additional features are NVM Set aware
- Indicate how the Namespace Attribute Changed event is cleared; additional cleanup and clarification.
- For Get Features command: move the completion queue info from 5.13.1 to 5.13.2
- Correct the name of the Write Atomicity Normal feature
- In the Error Count field of the Error Information Log Entry: add a requirement to wrap to 00000001h.
- In the Establish Context and Read Log Data field in Command Dword 10 of the Get Log Page command: deletes the requirement: no log data shall be returned if the number of dwords to be returned is set to zero
- In the Sanitize command: if Sanitize Sanitize Action set to 001b (i.e., Exit Failure Mode) when Sanitize is not in process and has not failed: require that the command shall complete with a status of Successful Completion and perform no other action
- In the Create I/O Submission Queue command, clarify which SQ is being referenced in several places
- In the Get Log Page command, clarify backward compatibility for NSID to access the controller
- SMART/Health log
- In the Critical Warning field, clarify that bit 3 refers to ALL of the media being placed in a read only mode
- In the Host Read Commands field: add that it also counts the number of Verify commands
- In the Supported Events Bitmap field of the Persistent Event Log, add that bit 223 means ‘TCG Defined’
- In Figure 221: Additional Hardware Error Information for correctable and uncorrectable PCIe errors: correct the byte offset of the end of the data structure
- In the NVMCAP field, corrected a mathematical relationship
- Clarify that namespace attachments to a secondary controller persist even if that secondary controller is offline
- In Figure 262: Namespace Management – Host Software Specified Fields: clarify footnote 1 to indicate that if the associate feature is not supported then the field is ignored by the controller
- Refer to the correct names of the LBA Status Information Attributes data structure and the Power State Descriptor data structure
- In the Write Uncorrectable command: change ‘be marked as invalid’ to ‘become uncorrectable’
- For PRCHK: clarify the requirements when PRCHK=1 for each Type 1, 2 and 3 protection
- In the Virtualization Enhancements feature, clarify that secondary controllers should not support privileged actions

Description of Specification Changes

Markup Conventions:

Black:	Unchanged (however, hot links are removed)
~~Red Strikethrough~~:	Deleted
Blue underscore:	New
Red Highlighted:	TBD values, anchors, and links to be inserted.
<Green Bracketed>:	Notes to editor

Specific changes

<Modify portions of Section 1.9 (References) as follows>

…

~~INCITS 501-2016, Information technology – Security Features for SCSI Commands (SFSC). Available from http://webstore.ansi.org.~~

…

TCG Storage Interface Interactions Specification (SIIS), Version 1.08 Revision 1.00. Available from http://www.trustedcomputinggroup.org.

…

<Modify portions of Figure 93: Offset E08h: PMRSTS – Persistent Memory Region Status >

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>11:9</td><td>RO</td><td>000b</td><td>Health Status (HSTS): If the Persistent Memory Region is ready, then this field indicates the health status of the Persistent Memory Region. This field is always cleared to 000b when the Persistent Memory Region is not ready.

The health status values are defined as:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Normal Operation: The Persistent Memory Region is operating normally.</td></tr>
<tr><td>001b</td><td>Restore Error: The Persistent Memory Region is operating normally and is persistent; however, the contents of the Persistent Memory Region may not have been restored correctly (i.e., may not contain the contents prior to the last power cycle, NVM Ssubsystem Rreset, Controller Level Reset, or Persistent Memory Region disable).</td></tr>
<tr><td>010b</td><td>Read Only: The Persistent Memory Region is read only. PCI Express memory write requests do not update the Persistent Memory Region. PCI Express memory read requests to the Persistent Memory Region return correct data.</td></tr>
<tr><td>011b</td><td>Unreliable: The Persistent Memory Region has become unreliable. PCI Express memory reads may return invalid data or generate poisoned PCI Express TLP(s). Persistent Memory Region memory writes may not update memory or may update memory with undefined data. The Persistent Memory Region may also have become non-persistent.</td></tr>
<tr><td>100b to 111b</td><td>Reserved</td></tr>
<tr><td></td><td></td></tr>
</table>
</td></tr>
</table>

<Modify Figure 106: Command Format – Admin and NVM Vendor Specific Commands (Optional)>

Bytes	Description
03:00	**Command Dword 0 (CDW0):** This field is common to all commands and is defined in Figure 104.
07:04	**Namespace Identifier (NSID):** This field indicates the namespace ID that this command applies to. If the namespace ID is not used for the command, then this field shall be cleared to 0h. Setting this value to FFFFFFFFh causes the command to be applied to all namespaces attached to ~~this controller~~ the controller processing the command, unless otherwise specified. The behavior of a controller in response to an inactive namespace ID for a vendor specific command is vendor specific. Specifying an invalid namespace ID in a command that uses the namespace ID shall cause the controller to abort the command with status Invalid Namespace or Format, unless otherwise specified.
15:08	Reserved
39:16	Refer to Figure 105 for the definition of these fields.
43:40	**Number of Dwords in Data Transfer (NDT):** This field indicates the number of dwords in the data transfer.
47:44	**Number of Dwords in Metadata Transfer (NDM):** This field indicates the number of dwords in the metadata transfer.
51:48	**Command Dword 12 (CDW12):** This field is command specific Dword 12.
55:52	**Command Dword 13 (CDW13):** This field is command specific Dword 13.
59:56	**Command Dword 14 (CDW14):** This field is command specific Dword 14.
63:60	**Command Dword 15 (CDW15):** This field is command specific Dword 15.

<Modify 4.5 Metadata Region (MR) >

4.5 Metadata Region (MR)

Metadata may be supported for a namespace as part of the logical block (creating an extended logical block which is a larger logical block that is exposed to the application) or metadata may be transferred as a separate buffer of data. The metadata shall not be split between the logical block and a separate metadata buffer. For writes, the metadata shall be written atomically with its associated logical block. Refer to section 8.2.

In the case where the namespace is formatted to transfer the metadata as a separate buffer of data, then the Metadata Region is used. In this case, the location of the Metadata Region is indicated by the Metadata Pointer within the command. The Metadata Pointer within the command shall be dword aligned.

The controller may support several physical formats of logical block size and associated metadata size. There may be performance differences between different physical formats. This is indicated as part of the Identify Namespace data structure.

If the namespace is formatted to use end-to-end data protection (refer to section 8.3), then the first eight bytes or last eight bytes of the metadata is used for protection information (specified as part of the Format NVM command ~~Format operation~~).

Modify Figure 126: Status Code – Generic Command Status Values as noted below>

Value	Description
15h	**Operation Denied:** The command was denied due to lack of access rights. Refer to the appropriate security specification (e.g., TCG Storage Interface Interactions S~~s~~pecification). For media access commands, the Access Denied status code should be used instead.

<Modify figure 129>

Figure 129: Status Code – Command Specific Status Values, NVM Command Set

Value	Description	Commands Affected
80h	Conflicting Attributes	Dataset Management, Read, Write
81h	Invalid Protection Information	Compare, Read, Verify, Write, Write Zeroes
82h	Attempted Write to Read Only Range	Dataset Management, Write, Write Uncorrectable, Write Zeroes, Flush, Format NVM
83h to BFh	Reserved	

<Modify Figure 131: Status Code – Media and Data Integrity Error Values, NVM Command Set>

Value	Description
80h	**Write Fault:** The write data could not be committed to the media.
81h	**Unrecovered Read Error:** The read data could not be recovered from the media.
82h	**End-to-end Guard Check Error:** The command was aborted due to an end-to-end guard check failure.
83h	**End-to-end Application Tag Check Error:** The command was aborted due to an end-to-end application tag check failure.
84h	**End-to-end Reference Tag Check Error:** The command was aborted due to an end-to-end reference tag check failure.
85h	**Compare Failure:** The command failed due to a miscompare during a Compare command.
86h	**Access Denied:** Access to the namespace and/or LBA range is denied due to lack of access rights. Refer to the appropriate security specification (e.g., TCG Storage Interface Interactions Sspecification).
87h	**Deallocated or Unwritten Logical Block:** The command failed due to an attempt to read from or verify an LBA range containing a deallocated or unwritten logical block.
88h to BFh	Reserved

<Modify section 4.9 (NVM Sets) as noted below>

The host determines the NVM Sets present and their attributes using the Identify command with CNS value of 04h to retrieve the NVM Set List (refer to Figure 250). For each NVM Set, the attributes include:
• an identifier associated with the NVM Set;
• the optimal size for writes to the NVM Set;
• the total capacity of the NVM Set; and
• the unallocated capacity for the NVM Set.

An NVM Set Identifier is a 16-bit value that specifies the NVM Set with which an action is associated. An NVM Set Identifier may be specified in NVM Set aware Admin commands (refer to Figure 134). An NVM Set Identifier value of 0h is reserved and is not a valid NVM Set Identifier. Unless otherwise specified, if the host specifies an NVM Set Identifier cleared to 0h for a command that requires an NVM Set Identifier, then that command shall ~~fail~~ abort with a status code of Invalid Field in Command.

Each NVM Set is associated with exactly one Endurance Group (refer to section 8.17).

…

<Modify figure 134 NVM Set Aware Admin commands >

Figure 134: NVM Set Aware Admin Commands

Admin Command	Details
Identify	• The Identify Namespace data structure includes the associated NVM Set Identifier. • The NVM Set List data structure includes attributes for each NVM Set.
Namespace Management	• The create action includes the NVM Set Identifier as a host specified field.
Get Features and Set Features	• The Read Recovery Level Feature specifies the associated NVM Set Identifier. • The Predictable Latency Mode Config Feature specifies the associated NVM Set Identifier. • The Predictable Latency Mode Window Feature specifies the associated NVM Set Identifier.

The host determines the NVM Sets present and their attributes using the Identify command with CNS value

<Modify Figure 147: Asynchronous Event Information – Notice as noted below>

Value	Description
00h	**Namespace Attribute Changed:** Indicates a change to one or both of the following: • the Identify Namespace data structure (refer to Figure 245) for one or more namespaces; or • the Namespace List returned when the Identify command is issued with the CNS field set to 02h. ~~The Identify Namespace data structure (refer to Figure 245) 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 to read the Identify Namespace data structures for each namespace to determine what has changed.~~ ~~Alternatively, host software may request the Changed Namespace List (Log Identifier 04h) (refer to section 5.14.1.4) to determine which namespaces in this controller have changed information in the Identify Namespace data structure since the last time the log page was read.~~ To clear this event, host software issues a Get Log Page command for the Changed Namespace List log page (Log Identifier 04h - refer to section 5.14.1.4) with the Retain Asynchronous Event bit cleared to '0'. A controller shall not send this event if: a) Namespace Utilization (refer to Figure 245) has changed, as this is a frequent event that does not require action by the host; b) the ANAGRPID field (refer to Figure 245) has changed; or c) capacity information (i.e., the NUSE field and the NVMCAP field) returned in the Identify Namespace data structure (refer to Figure 245) changed as a result of an ANA state change. 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 0h, or from 0h to a non-zero value.
…	…

<Modify Figure 151: Create I/O Completion Queue – Command Dword 11>

Bits	Description
31:16	**Interrupt Vector (IV):** This field indicates interrupt vector to use for this Completion Queue. This corresponds to the MSI-X or multiple message MSI vector to use. If using single message MSI or pin-based interrupts, then this field shall be cleared to 0h. In MSI-X, a maximum of 2,048 vectors are used. This value shall not be set to a value greater than the number of messages the controller supports (refer to MSICAP.MC.MME or MSIXCAP.MXC.TS). If the value is greater than the number of messages the controller supports, the controller should return an error of Invalid Interrupt Vector.
15:02	Reserved
01	**Interrupts Enabled (IEN):** If set to '1', then interrupts are enabled for this Completion Queue. If cleared to '0', then interrupts are disabled for this Completion Queue.
00	**Physically Contiguous (PC):** If set to '1', then the Completion Queue is physically contiguous and PRP Entry 1 (PRP1) is the address of a contiguous physical buffer. If cleared to '0', then the Completion Queue is not physically contiguous and PRP Entry 1 (PRP1) is a PRP List pointer. If the: • queue is located in the Controller Memory Buffer; • PC is cleared to '0'; and • CMBLOC.CQPDS is cleared to '0', then the controller shall abort ~~fail~~ the command with Invalid Use of Controller Memory Buffer status.

<modify section 5.4 Create I/O Submission Queue command>

5.4 Create I/O Submission Queue command

The Create I/O Submission Queue command is used to create I/O Submission Queues. The Admin Submission Queue is created by specifying its base address in the ASQ register. If a PRP List is provided to describe the SQ to be created, then the PRP List shall be maintained by host software at the same location in host physical memory and the values in the PRP List shall not be modified until the corresponding Delete I/O Submission Queue command for that ~~this~~ SQ is completed or the controller is reset. If the PRP List values are modified, the behavior is undefined.

<Modify Figure 155: Create I/O Submission Queue – Command Dword 11>

Bits	Description
…	**…**
00	**Physically Contiguous (PC):** If set to '1', then the Submission Queue is physically contiguous and PRP Entry 1 (PRP1) is the address of a contiguous physical buffer. If cleared to '0', then the Submission Queue is not physically contiguous and PRP Entry 1 (PRP1) is a PRP List pointer. If this bit is cleared to '0' and CAP.CQR is set to '1', the controller should return an error of Invalid Field in Command. If the: • queue is located in the Controller Memory Buffer; • PC is cleared to '0'; and • CMBLOC.CQPDS is cleared to '0', then the controller shall abort ~~fail~~ the command with Invalid Use of Controller Memory Buffer status.

<Modify section 5.5 Delete I/O Completion Queue command>

The Delete I/O Completion Queue command is used to delete an I/O Completion Queue. The Delete I/O Completion Queue command uses the Command Dword 10 field. All other command specific fields are reserved. After this command has completed, the PRP List that describes the Completion Queue may be deallocated by host software.

Host software shall ensure that any associated I/O Submission Queue is deleted prior to deleting a Completion Queue. If there are any associated I/O Submission Queues present, then the Delete I/O Completion Queue command shall abort ~~fail~~ with a status value of Invalid Queue Deletion.

Note: It is not possible to delete the Admin Completion Queue.

<Modify figure 159: remove extra whitespace in the 'value' column and Description column for value 0Ch>

Figure 159: Delete I/O Completion Queue – Command Specific Status Values

Value	Description
01h	**Invalid Queue Identifier:** The Queue Identifier specified in the command is invalid. This error is also indicated if the Admin Completion Queue identifier is specified.
0Ch	**Invalid Queue Deletion:** This error indicates that it is invalid to delete the I/O Completion Queue specified. The typical reason for this error condition is that there is an associated I/O Submission Queue that has not been deleted.

<Modify Figure 178: Firmware Image Download – Command Dword 10>

Bits	Description
31:00	**Number of Dwords (NUMD):** This field specifies the number of dwords to transfer for this portion of the firmware. This is a 0's based value. If the value specified in this field does not meet the requirement indicated by the FWUG field (refer to Figure 247), the firmware update may abort ~~fail~~ with status of Invalid Field in Command.

…

<Modify Figure 184: Get Features – Feature Identifiers>

Description	Section Defining Format of Attributes Returned
Arbitration	5.21.1.1
Power Management	5.21.1.2
LBA Range Type	5.21.1.3
Temperature Threshold	5.21.1.4
Error Recovery	5.21.1.5
Volatile Write Cache	5.21.1.6
Number of Queues	5.21.1.7
Interrupt Coalescing	5.21.1.8
Interrupt Vector Configuration	5.21.1.9
Write Atomicity Normal	5.21.1.10
Asynchronous Event Configuration	5.21.1.11
Autonomous Power State Transition	5.21.1.12

<Modify 5.13.1 Select field and 5.13.2 Command Completion>

5.13.1 Select field

A Select field cleared to 000b (i.e., current) returns the current operating attribute value for the Feature Identifier specified.

A Select field set to 001b (i.e., default) returns the default attribute value for the Feature Identifier specified.

A Select field set to 010b (i.e., saved) returns the last saved attribute value for the Feature Identifier specified (i.e., the last Set Features command completed without error, with the Save bit set to '1' for the Feature Identifier specified).

A Select field set to 011b (i.e., supported capabilities) returns the capabilities supported for this Feature Identifier. The capabilities supported are returned in Dword 0 of the completion entry of the Get Features command (refer to Figure NEW).

- ~~If Dword 0 bit 0 of the completion entry of the Get Features command is set to '1', then the Feature Identifier is saveable. If Dword 0 bit 0 of the completion entry of the Get Features command is cleared to '0', then the Feature Identifier is not saveable;~~
- ~~If Dword 0 bit 1 of the completion entry of the Get Features command is set to '1', then the Feature Identifier is namespace specific and settings are applied to individual namespaces. If Dword 0 bit 1 of the completion entry of the Get Features command is cleared to '0', then the Feature Identifier is not namespace specific and its settings apply to the entire controller; or~~
- ~~If Dword 0 bit 2 of the completion entry of the Get Features command is set to '1', then the Feature Identifier is changeable. If Dword 0 bit 2 of the completion entry of the Get Features command is cleared to '0', then the Feature Identifier is not changeable.~~

5.13.2 Command Completion

Upon completion of the Get Features command, the controller posts a completion queue entry to the Admin Completion Queue. If the Select field is not set to 11b, then ~~Depending on Feature Identifier field,~~ Dword 0 of the completion queue entry may contain feature-dependent information (refer to section 5.21.1).

If the Select field is set to 11b, then Figure NEW describes the contents of Dword 0 of the completion queue entry.

Figure NEW - Completion Queue Entry Dword 0 when Select=11b

Bits	**Description**
31:3	Reserved
2	**Changeable**: If set to '1', then the feature values are changeable. If cleared to '0', then the feature values are not changeable.
1	**NS Specific**: If set to '1', then the Feature Identifier is namespace specific and settings are applied to individual namespaces. If cleared to '0', then the Feature Identifier is not namespace specific and its settings apply to the entire controller.
0	**Saveable**: If set to '1', then the feature values are saveable. If cleared to '0', then the feature values are not saveable.

<Modify Figure 193: Get Log Page – Error Information Log Entry (Log Identifier 01h)>

Bytes	Description
07:00	**Error Count:** This is a 64-bit incrementing error count, indicating a unique identifier for this error. The error count starts at 1h, is incremented for each unique error log entry, and is retained across power off conditions. A value of 0h indicates an invalid entry; this value is used when there are lost entries or when there are fewer errors than the maximum number of entries the controller supports. If the value of this field is FFFFFFFFh, then the field shall be set to 1h when incremented (i.e., rolls over to 1h). Prior to NVMe 1.4, processing of incrementing beyond FFFFFFFFh is unspecified.

<Modify section 5.14.1.2>

This log page is used to provide SMART and general health information. The information provided is over the life of the controller and is retained across power cycles. To request the controller log page, the namespace identifier specified is FFFFFFFFh or 0h. For compatibility with implementations compliant with revision 1.4 of this specification and earlier, hosts should use a namespace identifier of FFFFFFFFh to request the controller log page. The controller may also support requesting the log page on a per namespace basis, as indicated by bit 0 of the LPA field in the Identify Controller data structure in Figure 247

…

<Modify Figure 194: Get Log Page – SMART / Health Information Log>

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>00</td><td>Critical Warning: This field indicates critical warnings for the state of the controller. Each bit corresponds to a critical warning type; multiple bits may be set to ‘1’. If a bit is cleared to ‘0’, then that critical warning does not apply. Critical warnings may result in an asynchronous event notification to the host. Bits in this field represent the current associated state and are not persistent.

<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>7:6</td><td>Reserved</td></tr>
<tr><td>5</td><td>If set to ‘1’, then the Persistent Memory Region has become read-only or unreliable (refer to section 4.8).</td></tr>
<tr><td>4</td><td>If set to ‘1’, then the volatile memory backup device has failed. This field is only valid if the controller has a volatile memory backup solution.</td></tr>
<tr><td>3</td><td>If set to ‘1’, then all of the media has been placed in read only mode. The controller shall not set this bit to '1' if the read-only condition on the media is a result of a change in the write protection state of a namespace (refer to section 8.19.1).</td></tr>
<tr><td>2</td><td>If set to ‘1’, then the NVM subsystem reliability has been degraded due to significant media related errors or any internal error that degrades NVM subsystem reliability.</td></tr>
<tr><td>1</td><td>If set to ‘1’, then a temperature is:
a) greater than or equal to an over temperature threshold; or
b) less than or equal to an under temperature threshold,
(refer to section 5.21.1.4).</td></tr>
<tr><td>0</td><td>If set to ‘1’, then the available spare capacity has fallen below the threshold.</td></tr>
</table></td></tr>
<tr><td>02:01</td><td>Composite Temperature: Contains a value corresponding to a temperature in ~~degrees Kelvin~~ Kelvins that represents the current composite temperature of the controller and namespace(s) associated with that controller. The manner in which this value is computed is implementation specific and may not represent the actual temperature of any physical point in the NVM subsystem. The value of this field may be used to trigger an asynchronous event (refer to section 5.21.1.4).

Warning and critical overheating composite temperature threshold values are reported by the WCTEMP and CCTEMP fields in the Identify Controller data structure in Figure 247.</td></tr>
<tr><td>…</td><td></td></tr>
</table>

<Modify 5.14.1.4 Changed Namespace List (Log Identifier 04h)>

This log page is used to describe namespaces attached to the ~~this~~ controller that have:

a) changed information in their Identify Namespace data structure since the last time the log page was read;
b) been added; and
c) been deleted.

The log page contains a Namespace List with up to 1,024 entries. If more than 1,024 namespaces have changed attributes since the last time the log page was read, the first entry in the log page shall be set to FFFFFFFFh and the remainder of the list shall be zero filled.

<Modify Figure 204: Get Log Page – Endurance Group Log (Log Identifier 09h)>

Bytes	Description
. . .	
111:96	**Host Read Commands:** Contains the number of read commands completed by all controllers in the NVM subsystem for the Endurance Group. For the NVM command set, this is the number of Compare commands, ~~and~~ Read commands, and Verify commands.
…	…

<Modify 5.14.1.13 Persistent Event Log (Log Identifier 0Dh)>

…

Events that affect multiple controllers (e.g., an NVM Ssubsystem Rreset) should be logged once by a controller selected by the vendor and not logged by any other controllers.

…

The controller should retain the persistent event log reporting context:

a) Until the controller processes:
   a) A Get Log Page command requesting the Persistent Event Log page with the Action field set to 02h (i.e., Release Context);
   b) An NVM Ssubsystem Rreset; or
   c) A Controller Level Reset;
   or
b) For a vendor specific time long enough to allow retrieval of the persistent event log page data.

…

<Modify Figure 210: Command Dword 10 – Log Specific Field>

<Note: the figure needs to be recreated because it was not created as a proper editable table>

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>09:08</td><td>Action: This field specifies the action the controller shall take during processing this Get Log Page command.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>Read Log Data: Return persistent event log page data starting at the address indicated the LPOU field and the LPOL field in the get Log Page command. If the controller does have a persistent event log reporting context, then the controller shall abort ~~fail~~ the comm with a status code of Command Sequence Error.</td></tr>
<tr><td>01b</td><td>Establish Context and Read Log Data: The controller shall:
a) determine the length of the persistent event log page data;
b) determine the set of events to report in the persistent event log page data; and
c) establish a persistent event log reporting context to store information describing the persistent event log data to be reported and track persistent event log page data accesses.

After establishing a persistent event log reporting context the controller shall return persistent event log page data starting at the address indicated by the LPOU field and the LPOL field in the Get Log Page command.

~~A number of dwords to return set to zero in the Get Log Page command indicates that no log data shall be returned.~~

If a persistent event log reporting context already exists, then the controller shall abort ~~f~~ the command with a status code of Command Sequence Error.</td></tr>
<tr><td>10b</td><td>Release Context: The controller shall release the persistent event log reporting cont if any. It is not an error if the controller does not have a persistent event log repor context.</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table></td></tr>
</table>

<Modify Figure 211: Get Log Page – Persistent Event Log (Log Identifier 0Dh)>

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>51:44</td><td>Power Cycle Count: Contains the number of power cycles for the ~~this~~ controller.</td></tr>
<tr><td>53:52</td><td>PCI Vendor ID (VID): This is the same value as reported in the Identify Controller data structure PCI Vendor ID field (i.e., bytes 01:00).</td></tr>
<tr><td>55:54</td><td>PCI Subsystem Vendor ID (SSVID): This is the same value as reported in the Identify Controller data structure PCI Subsystem Vendor ID field (i.e., bytes 03:02).</td></tr>
<tr><td>75:56</td><td>Serial Number (SN): This field contains the same value as reported in the Serial Number field of the Identify Controller data structure, bytes 23:04.</td></tr>
<tr><td>115:76</td><td>Model Number (MN): This field contains the same value as reported in the Model Number field of the Identify Controller data structure, bytes 63:24.</td></tr>
<tr><td>371:116</td><td>NVM Subsystem NVMe Qualified Name (SUBNQN): This field contains the same value as reported in the NVM Subsystem NVMe Qualified Name field of the Identify Controller data structure, bytes 1023:768. If the NVM Subsystem NVMe Qualified Name field of the Identify Controller data structure is not supported, then all bytes of this field shall be cleared to 0h.</td></tr>
<tr><td>479:372</td><td>Reserved</td></tr>
<tr><td>511:480</td><td>Supported Events Bitmap: This field contains a bitmap indicating support for the persistent event log events. Each bit in the bitmap corresponds to the value for the event type (refer to Figure 213) (e.g., bit 222 decimal, DEh, corresponds to event type value DEh, Vendor Specific Event). A bit set to ‘1’ indicates that the corresponding event is supported. A bit cleared to ‘0’ indicates that the corresponding event is not supported.
<table>
<tr><th>Bits</th><th>Definition</th><th>Reference</th></tr>
<tr><td>255: 224~~223~~</td><td>Reserved</td><td></td></tr>
<tr><td>223</td><td>TCG Defined</td><td>TCG Storage Interface Interactions Specification</td></tr>
<tr><td>222</td><td>Vendor Specific Event Supported</td><td>5.14.1.13.1.14</td></tr>
<tr><td>221:14</td><td>Reserved</td><td></td></tr>
<tr><td>13</td><td>Thermal Excursion Event Support</td><td>5.14.1.13.1.13</td></tr>
<tr><td>12</td><td>Telemetry Log Create Event Support</td><td>5.14.1.13.1.12</td></tr>
<tr><td>11</td><td>Set Feature Event Support</td><td>5.14.1.13.1.11</td></tr>
<tr><td>10</td><td>Sanitize Completion Event Support</td><td>5.14.1.13.1.10</td></tr>
<tr><td>09</td><td>Sanitize Start Event Support</td><td>5.14.1.13.1.9</td></tr>
<tr><td>08</td><td>Format NVM Completion ~~Even~~ Event Support</td><td>5.14.1.13.1.8</td></tr>
<tr><td>07</td><td>Format NVM Start Event Support</td><td>5.14.1.13.1.7</td></tr>
<tr><td>06</td><td>Change Namespace Event Support</td><td>5.14.1.13.1.6</td></tr>
<tr><td>05</td><td>NVM Subsystem Hardware Error Event Support</td><td>5.14.1.13.1.5</td></tr>
<tr><td>04</td><td>Power-on or Reset Event Supported</td><td>5.14.1.13.1.4</td></tr>
<tr><td>03</td><td>Timestamp Change Event Supported</td><td>5.14.1.13.1.3</td></tr>
<tr><td>02</td><td>Firmware Commit Event Supported</td><td>5.14.1.13.1.2</td></tr>
<tr><td>01</td><td>SMART / Health Log Snapshot Event Supported</td><td>5.14.1.13.1.1</td></tr>
<tr><td>00</td><td>Reserved</td><td></td></tr>
</table></td></tr>
<tr><td colspan="2">Persistent Event Log Events</td></tr>
<tr><td>(M-1)+512:512</td><td>Persistent Event 0: This field contains the first persistent event log entry (refer to Figure 212) where M is the length of this persistent event.</td></tr>
<tr><td>…</td><td>…</td></tr>
<tr><td>(TLL-1):(TLL-K)</td><td>Persistent Event N: This field contains the last persistent event log entry (refer to Figure 212) where K is the length of this persistent event and TLL is the size specified in the Total Log Length field.</td></tr>
</table>

<Modify Figure 212: Persistent Event Format>

Bytes	Description
…	
05:04	**Controller Identifier:** This field contains the NVM subsystem unique controller identifier for the controller that created this event. If the event is controller specific, then the event data is associated with that ~~this~~ controller. If the event is not controller specific, then this is the controller that the NVM subsystem selected for creating the event.
…	

<Modify Figure 216: Timestamp Change Event Format (Event Type 03h)>

Bytes	Description
07:00	**Previous Timestamp:** Contains a timestamp using the format Timestamp – Data Structure for Get Features as defined in Figure ~~300~~ 301 containing the timestamp for the time immediately before the timestamp was changed (i.e., the old timestamp).
15:08	**Milliseconds Since Reset:** Contains the time since the last Controller Level Reset reported in milliseconds.

<Modify 5.14.1.13.1.4 Power-on or Reset Event (Event Type 04h)>

A Power-on or Reset event shall be recorded in the Persistent Event Log when an NVM Subystem Reset ~~subsystem reset~~ (e.g., due to a power-on) or a Controller Level Reset is completed. The Power-on or Reset Event reports information about resets due to power-on or other events that cause resets (refer to section 7.3) followed by descriptors reporting information about the controller at the time the reset occurred, including timestamp values for all controllers for use in synchronization of timestamp values across controllers.

<Modify Figure 218: Controller Reset Information descriptor>

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>01:00</td><td>Controller ID: Contains the Controller ID for the controller with the timestamp in the Controller Timestamp field.</td></tr>
<tr><td>02</td><td>Firmware Activation: Contains a code indicating if this event triggered a firmware activation.
<table>
<tr><th>Code</th><th>Definition</th></tr>
<tr><td>00h</td><td>Indicates that this event did not trigger a firmware activation on the ~~this~~ controller.</td></tr>
<tr><td>01h</td><td>Indicates that new firmware was activated on the ~~this~~ controller due to this power on or reset.</td></tr>
<tr><td>02h</td><td>Indicates that an attempt to activate new firmware on the ~~this~~ controller due to this power-on or reset failed.</td></tr>
<tr><td>03h to FFh</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>03</td><td>Operation in Progress:

Bits 7:1 are reserved.

Bit 0: A value of ‘1’ indicates that a Format NVM command was in progress for a namespace attached to the ~~this~~ controller when this reset event occurred. A value of ‘0’ indicates that no Format NVM commands were in progress for any namespace attached to the ~~this~~ controller when this reset event occurred.</td></tr>
<tr><td></td><td>…</td></tr>
</table>

<Modify Figure 220: NVM Subsystem Hardware Error Event Codes>

Code	Description
00h	Reserved
01h	**PCIe Correctable Error:** Indicates that the NVM subsystem has detected that a PCIe correctable error occurred. Refer to Figure 221 for the format of the Additional Hardware Error Information field.
02h	**PCIe Uncorrectable Non fatal Error:** Indicates that the NVM subsystem has detected that a PCIe uncorrectable non-fatal error occurred. Refer to Figure 221 for the format of the Additional Hardware Error Information field.
03h	**PCIe Uncorrectable Fatal Error:** Indicates that the NVM subsystem has detected that a PCIe uncorrectable fatal error occurred. Refer to Figure 221 for the format of the Additional Hardware Error Information field.
04h	**PCIe Link Status Change:** Indicates that a change in the values reported in the PCI Express Link Status register (refer to section 2.5.8) have changed due to an attempt to correct unreliable link operation. The Additional Hardware Error Information field shall be set to the contents of the PCI Express Link Status register at the time of the event.
05h	**PCIe Link Not Active:** Indicates that the Data Link Control and Management State Machine (refer to PCI Express Base specification) has transitioned out of the DL_Active state without a corresponding event (e.g., without an indication from the host that the link is to be disabled). This NVM subsystem hardware error event does not contain additional hardware error information.
06h	**Critical Warning Condition:** Indicates the NVM subsystem has detected a condition that causes a bit in the Critical Warning field of the SMART / Health Information log (refer to section 5.14.1.2) to be set to ~~one~~ '1'. Bits in this field represent the associated state at the time of this event. The Additional Hardware Error Information field shall be set at the time of the event using the same format as is specified for the Critical Warning field of the SMART / Health Information log.
07h	**Endurance Group Critical Warning Condition:** Indicates that the NVM subsystem has detected a condition that causes a bit in the Critical Warning field of an Endurance Group Information log (refer to section 5.14.1.9) to be set to '1'. Bits in this field represent the associated state at the time of this event. The Additional Hardware Error Information field shall be four bytes long.

Bytes	Definition
0	Shall be set at the time of the event using the same format as is specified for the Critical Warning field of the Endurance Group Information log page.
1	Reserved
3:2	Shall be set to the Endurance Group Identifier for the associated endurance group.

<Modify Figure 221: Additional Hardware Error Information for correctable and uncorrectable PCIe errors>

Bytes	Value
01:00	**PCIe Device Status Register:** Contains the contents of the PCI Device Status Register (refer to the PCI Express specification) at the time of the event.

02	Bits 7:1 Reserved Bit 0 **PCIe AER Supported:** set to '1' indicates that PCIe AER (refer to the PCI Express specification) is supported and that the PCIe AER Error Status field, PCIe AER Error Mask field, PCIe AER Header Log Register field, and the PCIe AER TLP Prefix Log Register field is reported. Bit 0 cleared to '0' indicates that PCIe AER is not supported and that the PCIe AER Error Status field, PCIe AER Error Mask field, PCIe AER Header Log Register field, and PCIe AER TLP Prefix Log Register field is not reported (i.e., bytes 80:16 are not reported).
15:03	Reserved
31:16	**PCIe AER Error Status:** Contains the contents of: a) the PCIe AER Correctable Error Status Register (refer to section 2.6.5) at the time of the event if the error is a correctable error; or b) The PCIe AER Uncorrectable Error Status Register (refer to section 2.6.2), at the time of the event if the error is an uncorrectable error.
47:32	**PCIe AER Error Mask:** Contains the contents of a) the PCIe AER Correctable Error Mask Register (refer to section 2.6.6) at the time of the event if the error is a correctable error; or b) the PCIe AER Uncorrectable Error Mask Register (refer to section 2.6.3) at the time of the event if the error is an uncorrectable error.
63:48	**PCIe AER Header Log Register:** Contains the contents of the PCIe AER Header Log Register (refer to section 2.6.8), if supported, at the time of the event.
7980:64	**PCIe AER TLP Prefix Log Register:** Contains the contents of the PCIe AER TLP Prefix Log Register (refer to section 2.6.9), if supported, at the time of the event.

<modify 5.14.1.16.2 Sanitize Status (Log Identifier 81h)>

5.14.1.16.2 Sanitize Status (Log Identifier 81h)

The Sanitize Status log page is used to report sanitize operation time estimates and information about the most recent sanitize operation (refer to section 8.15). The Get Log Page command returns a data buffer containing a log page formatted as defined in Figure 238. This log page shall be retained across power cycles and resets. This log page shall contain valid data whenever CSTS.RDY is set to '1'.

If the Sanitize Capabilities (SANICAP) field in the Identify Controller data structure is not cleared to 0h (i.e., the Sanitize command is supported), then this log page shall be supported. If the Sanitize Capabilities field in the Identify Controller data structure is cleared to 0h, then this log page is reserved.

…

<Modify 5.15.2.1 Identify Namespace data structure (CNS 00h)>

If the Namespace Identifier (NSID) field specifies an active NSID, then the Identify Namespace data structure (refer to Figure 245) is returned to the host for that specified namespace. If that specified namespace is an inactive NSID, then the controller returns a zero filled data structure.

If the controller supports the Namespace Management capability (refer to section 8.12) and the NSID field is set to FFFFFFFFh, then the controller returns an Identify Namespace data structure that specifies capabilities that are common across namespaces for the ~~this~~ controller. If the controller does not support the Namespace Management capability and the NSID field is set to FFFFFFFFh, then the controller shall abort ~~fail~~ the command with a status code of Invalid Namespace or Format

<Modify Figure 245: Identify – Identify Namespace Data Structure, NVM Command Set Specific>

Bytes	O/M [1]	Description

…		
63:48	O	**NVM Capacity (NVMCAP):** This field indicates the total size of the NVM allocated to this namespace. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to section 8.12) is supported. Note: This field may not correspond to the logical block size multiplied by the Namespace Size field. Due to thin provisioning or other settings (e.g., endurance), this field may be larger or smaller than ~~than~~ the product of the logical block size and the Namespace Size reported. If the controller supports Asymmetric Namespace Access Reporting (refer to the CMIC field), and the relationship between the controller and the namespace is in the ANA Inaccessible state (refer to section 8.20.3.3) or the ANA Persistent Loss state (refer to section 8.20.3.4), then this field shall be cleared to 0h.
…		

<Modify 5.15.2.2 Identify Controller data structure (CNS 01h)>

The Identify Controller data structure (refer to Figure 247) is returned to the host for the ~~this~~ controller.

<Modify Figure 247: Identify – Identify Controller Data Structure>

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td>315:312</td><td>O</td><td>Replay Protected Memory Block Support (RPMBS): This field indicates if the controller supports one or more Replay Protected Memory Blocks (RPMBs) and the capabilities. Refer to section 8.10.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:24</td><td>Access Size: If the Number of RPMB Units field is non-zero, then this field indicates the maximum number of 512B units of data that may be read or written per RPMB access by Security Send or Security Receive commands for the ~~this~~ controller. This is a 0’s based value. A value of 0h indicates support for one unit of 512B of data.

If the Number of RPMB Units field is 0h, then this field shall be ignored.</td></tr>
<tr><td>23:16</td><td>Total Size: If the Number of RPMB Units field is non-zero, then this field indicates the number of 128 KiB units of data in each RPMB supported in the controller. This is a 0’s based value. A value of 0h indicates support for one unit of 128 KiB of data.

If the Number of RPMB Units field is 0h, this field shall be ignored.</td></tr>
<tr><td>15:06</td><td>Reserved</td></tr>
<tr><td>05:03</td><td>Authentication Method: This field indicates the authentication method used to access all RPMBs in the controller. The values for this field are:
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>HMAC SHA-256 (refer to RFC 6234)</td></tr>
<tr><td>001b to 111b</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>02:00</td><td>Number of RPMB Units: This field indicates the number of RPMB targets the controller supports. All RPMB targets supported shall have the same capabilities as defined in the RPMBS field. A value of 0h indicates the controller does not support Replay Protected Memory Blocks. If this value is non-zero, then the controller shall support the Security Send and Security Receive commands.</td></tr>
</table></td></tr>
<tr><td>317:316</td><td>O</td><td>Extended Device Self-test Time (EDSTT): If the Device Self-test command is supported, then this field indicates the nominal amount of time in one minute units that the controller takes to complete an extended device self-test operation when in power state 0. If the Device Self-test command is not supported, then this field is reserved.</td></tr>
</table>

318	O	**Device Self-test Options (DSTO)**: This field indicates the optional Device Self-test command or operation behaviors supported by the controller or NVM subsystem. Bits 7:1 are reserved. Bit 0 if set to '1', then the NVM subsystem supports only one device self-test operation in progress at a time. If cleared to '0', then the NVM subsystem supports one device self-test operation per controller at a time.
319	M	**Firmware Update Granularity (FWUG):** This field indicates the granularity and alignment requirement of the firmware image being updated by the Firmware Image Download command (refer to section 5.12). If the values specified in the NUMD field or the OFST field in the Firmware Image Download command do not conform to this granularity and alignment requirement, then the firmware update may abort ~~fail~~ with status of Invalid Field in Command. For the broadest interoperability with host software, it is recommended that the controller set this value to the lowest value possible. The value is reported in 4 KiB units (e.g., 1h corresponds to 4 KiB, 2h corresponds to 8 KiB). A value of 0h indicates that no information on granularity is provided. A value of FFh indicates there is no restriction (i.e., any granularity and alignment in dwords is allowed)
…		
351:348	O	**Number of ANA Group Identifiers (NANAGRPID):** This field indicates the number of ANA groups supported by the ~~this~~ controller. If the controller supports Asymmetric Namespace Access Reporting (refer to the CMIC field), then this field shall be set to a non-zero value that is less than or equal to the ANAGRPMAX value. If the controller does not support Asymmetric Namespace Access Reporting, then this field shall be cleared to 0h
…		
533:532		**Atomic Compare & Write Unit (ACWU):** This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format for a Compare and Write fused operation. If a specific namespace guarantees a larger size than is reported in this field, then the Atomic Compare & Write Unit size for that namespace is reported in the NACWU field in the Identify Namespace data structure. Refer to section 6.4. This field shall be supported if the Compare and Write fused command is supported. This field is specified in logical blocks and is a 0's based value. If a Compare and Write is submitted that requests a transfer size larger than this value, then the controller may abort ~~fail~~ the command with a status code of Invalid Field in Command. If Compare and Write is not a supported fused command, then this field shall be 0h.
…		

<Modify 5.15.2.7 Identify Namespace data structure for an Allocated Namespace ID (CNS 11h)>

The Identify Namespace data structure (refer to Figure 245) is returned to the host for the namespace specified in the Namespace Identifier (NSID) field if it is an allocated NSID. If the specified namespace is an unallocated NSID then the controller returns a zero filled data structure.

If the specified namespace is an invalid NSID then the controller shall abort ~~fail~~ the command with a status code of Invalid Namespace or Format. If the NSID field is set to FFFFFFFFh then the controller should abort ~~fail~~ the command with a status code of Invalid Namespace or Format.

<Modify 5.15.2.8 Namespace Attached Controller list (CNS 12h)>

A Controller List (refer to section 4.11) of up to 2,047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers that are attached to the namespace specified in the Namespace Identifier (NSID) field. If the NSID field is set to FFFFFFFFh, then the controller should abort ~~fail~~ the command with a status code

of Invalid Field in Command.

<Modify 5.19 Namespace Attachment command>

The Namespace Attachment command is used to attach and detach controllers from a namespace. The attach and detach operations are persistent across all reset events. Namespace attach and detach operations are persistent across Virtualization Management commands that set a secondary controller offline.

If the Namespace Attachment command is supported, then the Namespace Management command (refer to section 5.20) shall also be supported.

The Namespace Attachment command uses the Data Pointer and Command Dword 10 fields. All other command specific fields are reserved.

<Modify figure 262: add footnote to NVM Set Id, generalize footnote>

Figure 262: Namespace Management – Host Software Specified Fields

Bytes	Description	Host Specified
07:00	Namespace Size (NSZE)	Yes
15:08	Namespace Capacity (NCAP)	Yes
25:16	Reserved	
26	Formatted LBA Size (FLBAS)	Yes
28:27	Reserved	
29	End-to-end Data Protection Type Settings (DPS)	Yes
30	Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC)	Yes
91:31	Reserved	
95:92	ANA Group Identifier (ANAGRPID) [1]	Yes
99:96	Reserved	
101:100	NVM Set Identifier (NVMSETID) [1]	Yes
383:102	Reserved	

Notes:
1. A value of 0h specifies that the controller determines the value to use (refer to section 8.12). If the associated feature is not supported, then this field is ignored by the controller.

<Modify Figure 266: Namespace Management – Command Specific Status Values>

Value	Description
0Ah	**Invalid Format:** The LBA Format specified is not supported. This may be due to various conditions, including: 1) specifying an invalid LBA Format number; 2) enabling protection information when there is not sufficient metadata per LBA; 3) the specified format is not available in the current configuration; or 4) invalid security state (refer to TCG Storage Interface Interactions Sspecification).
…	

<Modify 5.21.1 Feature Specific Information>

Figure 271 defines the Features that may be configured with a Set Features command and retrieved with a Get Features command. Figure 272 defines Features that are specific to the NVM Command Set. Refer to section 7.1 for mandatory, optional, and prohibited features for the various controller types. Some Features utilize a memory buffer to configure or return attributes for a Feature, whereas others only utilize a dword in the command or completion queue entry. Feature values that are not persistent across power cycles and

resets are restored to their default values as part of a Controller Level Reset ~~controller reset operation~~. For more information on Features, including default value definitions, saved ~~saveable~~ value definitions, and current value definitions, refer to section 7.8.

There may be commands in execution when a Feature is changed. The new settings may or may not apply to commands already submitted for execution when the Feature is changed. Any commands submitted to a Submission Queue after a Set Features command is successfully completed shall utilize the new settings for the associated Feature. To ensure that a Features values apply to all subsequent commands, the host should allow commands being processed to complete prior to issuing the Set Features command.

If the controller does not support a changeable value for a Feature (e.g., the Feature is not changeable), and a Set Feature command for that Feature is processed, then if that command specifies a Feature value that:

- is not the same as the existing value for that Feature, then the controller shall abort that command with a status code of Feature Not Changeable; and
- is the same as the existing value for that Feature, then the controller may:
  - complete that command successfully; or
  - abort that command with a status code of Feature Not Changeable.

…

<Modify 5.21.1.1 Arbitration (Feature Identifier 01h)>

This Feature controls command arbitration. Refer to section 4.13 for command arbitration details. The attributes are specified ~~indicated~~ in Command Dword 11.

…

<Modify 5.21.1.6 Volatile Write Cache (Feature Identifier 06h), (Optional)>

This Feature controls the volatile write cache, if present, on the controller. If a volatile write cache is present (refer to the VWC field in Figure 247), then this feature shall be supported. The attributes are specified ~~indicated~~ in Command Dword 11.

Note: If the controller is able to guarantee that data present in a write cache is written to non-volatile media on loss of power, then that write cache is considered non-volatile and this feature does not apply to that write cache.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 281 are returned in Dword 0 of the completion queue entry for that command.

If a volatile write cache is not present, then a Set Features command specifying the Volatile Write Cache feature identifier shall abort ~~fail~~ with Invalid Field in Command status, and a Get Features command specifying the Volatile Write Cache feature identifier should abort ~~fail~~ with Invalid Field in Command status.

Figure 281: Volatile Write Cache – Command Dword 11

Bits	Description
31:01	Reserved
00	**Volatile Write Cache Enable (WCE):** If set to '1', then the volatile write cache is enabled. If cleared to '0', then the volatile write cache is disabled.

…

<Modify 5.21.1.7 Number of Queues (Feature Identifier 07h)>

This Feature indicates the number of queues that the host requests for the controller processing the command ~~this controller~~. This feature shall only be issued during initialization prior to creation of any I/O Submission and/or Completion Queues. If a Set Features command is issued for this feature after creation of any I/O Submission and/or I/O Completion Queues, then the Set Features command shall abort ~~fail~~ with

status code of Command Sequence Error. The controller shall not change the value allocated between resets. For a Set Features command, the attributes are specified ~~indicated~~ in Command Dword 11 (refer to Figure 282). For a Get Features command, Dword 11 is ignored.

If a Set Features or Get Features command is submitted for this Feature, the attributes specified in Figure 283 are returned in Dword 0 of the completion queue entry for that command.

…

<Modify 5.21.1.10 Write Atomicity Normal (Feature Identifier 0Ah)>

This Feature controls the operation of the AWUN and NAWUN parameters (refer to section 6.4). The attributes are specified ~~indicated~~ in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 286 are returned in Dword 0 of the completion queue entry for that command.

….

<Modify 5.21.1.11 Asynchronous Event Configuration (Feature Identifier 0Bh)>

This Feature controls the events that trigger an asynchronous event notification to the host. This Feature may be used to disable reporting events in the case of a persistent condition (refer to section 5.2). If the condition for an event is true when the corresponding notice is enabled, then an event is sent to the host. The attributes are specified ~~indicated~~ in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 287 are returned in Dword 0 of the completion queue entry for that command.

…

<Modify 5.21.1.13 Host Memory Buffer (Feature Identifier 0Dh), (Optional)>

This Feature controls the Host Memory Buffer. The attributes are specified ~~indicated~~ in Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15.

The Host Memory Buffer feature provides a mechanism for the host to allocate a portion of host memory for the exclusive use of the controller. After a successful completion of a Set Features command enabling the host memory buffer, the host shall not write to:

a) The Host Memory Descriptor List (refer to Figure 296); and
b) the associated host memory region (i.e., the memory regions described by the Host Memory Descriptor List),

until the host memory buffer has been disabled.

If the host memory buffer is enabled, then a Set Features command to enable the host memory buffer (i.e., the EHM bit (refer to Figure 291) set to '1') shall abort ~~fail~~ with a status code of Command Sequence Error.

If the host memory buffer is not enabled, then a Set Features command to disable the host memory buffer (i.e., the EHM bit (refer to Figure 291) cleared to '0') shall succeed without taking any action.

…

<Modify section 5.21.1.14>

5.21.1.14 Timestamp (Feature Identifier 0Eh), (Optional)

The Timestamp feature enables the host to set a timestamp value in the controller. A controller indicates support for the Timestamp feature through the Optional NVM Command Support (ONCS) field in the Identify

Controller data structure. The Timestamp value (refer to Figure 300) in a Set Features command sets a timestamp value in the controller. After the current value for this feature is set, the controller updates that value as time passes. A Get Features command that requests the current value reports the timestamp value in the controller at the time the Get Features command is processed (e.g., the value set with a Set Features command for the current value plus the elapsed time since being set).

Note: If the Timestamp feature is saveable (refer to figure NEW) ~~supports a saveable value~~ and the host saves a value~~sets a saveable value~~, then the timestamp value restored after a subsequent power on or reset event is the value that was saved (refer to section 7.8). As a result, the timestamp may appear to move backwards in time.

…

<Modify 5.21.1.15 Keep Alive Timer (Feature Identifier 0Fh)>

This Feature controls the Keep Alive Timer. Refer to section 7.12 for Keep Alive details. The attributes are specified ~~indicated~~ in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 302 are returned in Dword 0 of the completion queue entry for that command.

…

<Modify Figure 303: HCTM – Command Dword 11>

Bits	Description
31:16	**Thermal Management Temperature 1 (TMT1):** This field specifies the temperature, in Kelvins ~~degrees Kelvin~~, when the controller begins to transition to lower power active power states or performs vendor specific thermal management actions while minimizing the impact on performance (e.g., light throttling) in order to attempt to reduce the Composite Temperature. A value cleared to 0h, specifies that this part of the feature shall be disabled. The range of values that are supported by the controller are indicated in the Minimum Thermal Management Temperature field and Maximum Thermal Management Temperature field in the Identify Controller data structure in Figure 247. If the host attempts to set this field to a value less than the value contained in the Minimum Thermal Management Temperature field or greater than the value contained in the Maximum Thermal Management Temperature field in the Identify Controller data structure in Figure 247, then the command shall abort ~~fail~~ with a status code of Invalid Field in Command. If the host attempts to set this field to a value greater than or equal to the value contained in the Thermal Management Temperature 2 field, if non-zero, then the command shall abort ~~fail~~ with a status code of Invalid Field in Command
15:00	**Thermal Management Temperature 2 (TMT2):** This field specifies the temperature, in Kelvins ~~degrees Kelvin~~, when the controller begins to transition to lower power active power states or perform vendor specific thermal management actions regardless of the impact on performance (e.g., heavy throttling) in order to attempt to reduce the Composite Temperature. A value cleared to 0h, specifies that this part of the feature shall be disabled. The range of values that are supported by the controller are indicated in the Minimum Thermal Management Temperature field and Maximum Thermal Management Temperature field in the Identify Controller data structure in Figure 247. If the host attempts to set this field to a value less than the value contained in the Minimum Thermal Management Temperature field or greater than the value contained in the Maximum Thermal Management Temperature field in the Identify Controller data structure in Figure 247, then the command shall abort ~~fail~~ with a status code of Invalid Field in Command. If the host attempts to set this field to a non-zero value less than or equal to the value contained in the Thermal Management Temperature 1 field, then the command shall abort ~~fail~~ with a status code of Invalid Field in Command

<Modify 5.21.1.18 Read Recovery Level Config (Feature Identifier 12h)>

This Feature is used to configure the Read Recovery Level (refer to section 8.16). The attributes are specified ~~indicated~~ in Command Dword 11 and Command Dword 12. Modifying the Read Recovery Level has no effect on the data contained in any associated namespace.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 306 are returned in Dword 0 of the completion queue entry for that command.

….

<Modify 5.21.1.19 Predictable Latency Mode Config (Feature Identifier 13h)>

This Feature configures an NVM Set to use Predictable Latency Mode, including warning event thresholds. Predictable Latency Mode and events are disabled by default. The attributes are specified ~~indicated~~ in Command Dword 11, Command Dword 12, and the Deterministic Threshold Configuration data structure.

…

<Modify 5.21.1.21 LBA Status Information Attributes (Feature Identifier 15h)>

The LBA Status Information Poll Interval (LSIPI) (refer to Figure 312) 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 312) 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 the host may use a polled method 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 value of the LBA Status Information Report Interval in Command Dword 11 (refer to Figure 312).

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 supported by the controller 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.

…

<Modify 5.21.1.22>

5.21.1.22 Host Behavior Support (Feature Identifier 16h)

This Feature enables use of controller functionality that is associated with and depends upon specific host behavior that may or may not be supported by all hosts. A controller does not use such functionality unless the host has indicated that the host supports the specific host behavior upon which the functionality depends. The host indicates that support to the controller by setting a field in this Feature. That host action enables controller use of the associated functionality with that host. A controller shall not use functionality with a host that has not indicated support for the associated specific host behavior upon which that controller functionality depends. The attributes in Figure 313 are transferred in the data buffer.

For example, the Command Interrupted status code is associated with and depends upon the specific host behavior that the host is expected to retry commands that are aborted with that status code. That command retry behavior may or may not be supported by all hosts (e.g., hosts based on versions of NVMe 1.3 and earlier are unlikely to retry commands aborted with the Command Interrupted status code as that status code was introduced after NVMe 1.3). A host that supports that command retry behavior indicates its support to the controller by setting a field to 1h in the Host Behavior Support Feature. Setting that field to 1h enables controller use of the Command Interrupted status code, with the result that this status code is used only with hosts that have indicated support for the associated command retry behavior.

This Feature is not saveable (refer to figure NEW). ~~Controllers shall not support a saveable value for this Feature, as host reboot with different host software could cause the contents of this Feature to become incorrect.~~ The default value of this Feature shall be all bytes cleared to 0h.

…

<Modify 5.21.1.23 Sanitize Config (Feature Identifier 17h), (Optional)>

This Feature controls behavior of the Sanitize command and sanitize operations. The scope of this Feature is the NVM subsystem.

The attributes are specified ~~indicated~~ in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 314 are returned in Dword 0 of the completion queue entry for that command.

If this Feature is not saveable (refer to figure NEW) ~~Dword 0 bit 0 of the completion entry of the Get Features command with SEL set to 011b (i.e., Supported Capabilities) for this Feature Identifier is cleared to '0', (i.e., not saveable, refer to section 7.8)~~, then the default value of the NODRM attribute shall be cleared to '0' (i.e., No-Deallocate Error Response Mode).

If the capabilities of the Sanitize Config Feature Identifier are both changeable and saveable (refer to section 7.8), then the host is able to configure this Feature when initially provisioning a device.

…

<Modify 5.21.1.24 Endurance Group Event Configuration (Feature Identifier 18h), (Optional)>

This Feature controls the events that trigger adding an Endurance Group Event Aggregate Log Change Notices event to the Endurance Group Event Aggregate log. This Feature may be used to disable reporting events in the case of a persistent condition (refer to section 5.2). If the condition for an event is true when the corresponding notice is enabled, then an event is sent to the host. The attributes are specified ~~indicated~~ in Command Dword 11.

If a Get Features command is submitted for this Feature, the Endurance Group Critical Warnings field in Command Dword 11 is not used and the attributes specified in Figure 315 are returned in Dword 0 of the completion queue entry for that command.

…

<Modify 5.21.1.25 Software Progress Marker (Feature Identifier 80h), (Optional) – NVM Command Set Specific>

This Feature is a software progress marker. The software progress marker is persistent across power states. For additional details, refer to section 7.6.1.1. This information may be used to indicate to an OS software driver whether there have been issues with the OS successfully loading. The attributes are specified ~~indicated~~ in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 316 are returned in Dword 0 of the completion queue entry for that command.

…

<Modify 5.21.1.27 Reservation Notification Mask (Feature Identifier 82h), (Optional[2])>

This Feature controls the masking of reservation notifications on a per namespace basis. A Reservation Notification log page is created whenever a reservation notification occurs on a namespace and the corresponding reservation notification type is not masked on that namespace by this Feature. If reservations are supported by the controller, then this Feature shall be supported. The attributes are specified ~~indicated~~ in Command Dword 11.

…

<Modify 5.21.1.28 Reservation Persistence (Feature Identifier 83h), (Optional[3])>

Each namespace that supports reservations has a Persist Through Power Loss (PTPL) state that may be modified using either a Set Features command or a Reservation Register command (refer to section 6.11). The Reservation Persistence feature attributes are specified ~~indicated~~ in Command Dword 11.

The PTPL state is contained in the Reservation Persistence Feature that is namespace specific. A Set Features command that uses the namespace ID FFFFFFFFh modifies the PTPL state associated with all namespaces that are attached to the controller and that support PTPL (i.e., support reservations). A Set Features command that uses a valid namespace ID other than FFFFFFFFh and corresponds to a namespace that supports reservations, modifies the PTPL state for that namespace. A Get Features command that uses a namespace ID of FFFFFFFFh should be aborted with status Invalid Field in Command. A Get Features command that uses a valid namespace ID other than FFFFFFFFh and corresponds to a namespace that supports PTPL, returns the PTPL state for that namespace. If a Set Features command or a Get Features command using a namespace ID other than FFFFFFFFh attempts to access the PTPL state for a namespace that does not support this Feature Identifier, then the command is aborted with status Invalid Field in Command.

This Feature should not be saveable (refer to figure NEW) ~~support a saveable value~~. If this Feature is ~~a~~ saveable ~~value is supported for this Feature,~~ then the host should set the current value and the saved ~~saveable~~ value to the same value.

If a Get Features command successfully completes for this Feature Identifier, the attributes specified in Figure 320 are returned in Dword 0 of the completion queue entry for that command

…

<Modify 5.21.1.29 Namespace Write Protection Config (Feature Identifier 84h)>

This Feature is used by the host to configure the namespace write protection state or to determine the write protection state of a namespace. Refer to section 8.19 for definition and behaviors of the namespace write protection states. The settings are specified in Command Dword 11.

This Feature is not saveable (refer to figure NEW ~~section 7.8~~). There is no default value for this Feature; the value of the Feature after a power cycle or a Controller Level Reset is determined by the write protection state of the namespace prior to the power cycle or Controller Level Reset, except for the Write Protect Until Power Cycle write protection state (refer to section 8.19).

If a Get Features command is submitted for this Feature, the attributes specified in Figure 321 are returned in Dword 0 of the completion queue entry for that command.

Figure 321: Write Protection – Command Dword 11

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:03</td><td>Reserved</td></tr>
<tr><td>02:00</td><td>Write Protection State: This field specifies the write protection state of the specified namespace.

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>No Write Protect</td></tr>
<tr><td>001b</td><td>Write Protect</td></tr>
<tr><td>010b</td><td>Write Protect Until Power Cycle</td></tr>
<tr><td>011b</td><td>Permanent Write Protect</td></tr>
<tr><td>100b to 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

If a Set Features command attempts to change the namespace write protection state of a namespace that is in the Write Protect Until Power Cycle state or the Permanent Write Protect state, then the command shall abort ~~fail~~ with a status code of Feature Not Changeable.

If a Set Features command attempts to change the namespace write protection state of a namespace to the Write Protect Until Power Cycle state and bit 0 of the of the Write Protection Authentication Control field is cleared to '0', then the command shall abort ~~fail~~ with a status code of Feature Not Changeable.

If a Set Features command changes the namespace to a write protected state, then the controller shall commit all volatile write cache data and metadata associated with the specified namespace to non-volatile media as part of transitioning to the write protected state.

…

<Modify 5.23 Format NVM command – NVM Command Set Specific>

…

The Format NVM command shall fail if the controller is in an invalid security state (refer to the appropriate security specification, e.g., TCG Storage Interface Interactions S~~s~~pecification). The Format NVM command may fail if there are outstanding I/O commands to the namespace specified to be formatted. I/O commands for a namespace that has a Format NVM command in progress may be aborted and if aborted, the controller should return a status code of Format in Progress.

…

<Modify 5.23.1: change from an ordered list to an unordered list>

5.23.1 Command Completion

A completion queue entry is posted to the Admin Completion Queue when the NVM media format is complete. Format NVM command specific status values are defined in Figure 329.

Figure 329: Format NVM – Command Specific Status Values

Value	Description
0Ah	**Invalid Format:** The format specified is invalid. This may be due to various conditions, including: ~~1) specifying an invalid LBA Format number;~~ ~~2) enabling protection information when there is not sufficient metadata per LBA;~~ ~~3) the specified format is not available in the current configuration; or~~ ~~4) invalid security state (refer to TCG Storage Interface Interactions specification), etc.~~ a) specifying an invalid LBA Format number; b) enabling protection information when there is not sufficient metadata per LBA; c) the specified format is not available in the current configuration; or d) invalid security state (refer to TCG Storage Interface Interactions Specification), etc.

<Modify 5.24 Sanitize >

5.24 Sanitize command – NVM Command Set Specific

The Sanitize command is used to start a sanitize operation or to recover from a previously failed sanitize operation. The sanitize operation types that may be supported are Block Erase, Crypto Erase, and Overwrite. All sanitize operations are processed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation). Refer to section 8.15 for details on the sanitize operation.

…

If a sanitize operation is not in progress and the most recent sanitize operation did not fail, then a Sanitize command with a Sanitize Action set to 001b (i.e., Exit Failure Mode) shall complete with a status of Successful Completion and perform no other action.

While a sanitize operation is in progress, all controllers in the NVM subsystem shall abort any command not allowed during a sanitize operation with a status of Sanitize In Progress (refer to section 8.15.1) and the Persistent Memory Region shall behave as described in section 8.15.1.

After a sanitize operation fails, all controllers in the NVM subsystem shall abort any command not allowed during a sanitize operation with a status of Sanitize Failed (refer to section 8.15.1) and the Persistent Memory Region shall behave as described in section 8.15.1 until a subsequent sanitize operation is started or successful recovery from the failed sanitize operation occurs.

If the most recent failed sanitize operation was started in unrestricted completion mode (i.e., the AUSE bit was set to ‘1’ in the Sanitize command), failure recovery requires the host to issue a subsequent Sanitize command in restricted or unrestricted completion mode or to issue a subsequent Sanitize command with the Exit Failure Mode action.

If the most recent failed sanitize operation was started in restricted completion mode (i.e., the AUSE bit was cleared to ‘0’ in the Sanitize command), failure recovery requires the host to issue a subsequent Sanitize command in restricted completion mode. In the case of a sanitize operation failure in restricted completion mode, before starting another sanitize operation:

- any subsequent Sanitize command issued with the Exit Failure Mode action shall be aborted with a status of Sanitize Failed; and
- any Sanitize command issued in unrestricted completion mode shall be aborted with a status of Sanitize Failed.

…

<modify sections 5.25 to change SFSC to SPC-4>

5.25 Security Receive command – NVM Command Set Specific

The Security Receive command transfers the status and data result of one or more Security Send commands that were previously submitted to the controller.

The association between a Security Receive command and previous Security Send commands is dependent on the Security Protocol. The format of the data to be transferred is dependent on the Security Protocol. Refer to SPC-4 ~~SFSC~~ for Security Protocol details.

Each Security Receive command returns the appropriate data corresponding to a Security Send command as defined by the rules of the Security Protocol. The Security Receive command data may not be retained if there is a loss of communication between the controller and host, or if a Controller Level Reset occurs.

The fields used are Data Pointer, Command Dword 10, and Command Dword 11 fields. All other command specific fields are reserved.

Figure 333: Security Receive – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 105 for the definition of this field.

Figure 334: Security Receive – Command Dword 10

Bits	Description
31:24	**Security Protocol (SECP):** This field specifies the security protocol as defined in SPC-4 ~~SFSC~~. The controller shall abort the command with status of Invalid Field in Command if an unsupported value
23:16	**SP Specific 1 (SPSP1):** The value of this field contains bits 15:08 of the Security Protocol Specific field as defined in SPC-4 ~~SFSC~~.
15:08	**SP Specific 0 (SPSP0):** The value of this field contains bits 07:00 of the Security Protocol Specific field as defined in SPC-4 ~~SFSC~~.
07:00	**NVMe Security Specific Field (NSSF):** Refer to Figure 336 for definition of this field for Security Protocol EAh. For all other Security Protocols this field is reserved.

Figure 335: Security Receive – Command Dword 11

Bits	Description
31:00	**Allocation Length (AL):** The value of this field is specific to the Security Protocol In command with the INC_512 field cleared to 0h as defined in SPC-4 ~~SFSC where INC_512 = 0~~.

5.25.1 Command Completion

If the command is completed, then the controller shall post a completion queue entry to the Admin Completion Queue indicating the status for the command.

5.25.2 Security Protocol 00h

A Security Receive command with the Security Protocol field cleared to 00h shall return information about the security protocols supported by the controller. This command is used in the security discovery process and is not associated with a Security Send command. Refer to SPC-4 ~~SFSC~~ for the details of Security Protocol 00h and the SP Specific field.

5.25.3 Security Protocol EAh

Security Protocol EAh is assigned for NVMe interface use (refer to ACS-4). This protocol may be used in Security Receive and Security Send commands. The specific usage type is defined by the Security Protocol Specific Field defined in Figure 336.

Figure 336: Security Protocol EAh – Security Protocol Specific Field Values

SP Specific (SPSP) Value	Description	NVMe Security Specific Field (NSSF) Definition
0001h	Replay Protected Memory Block	RPMB Target
0002h to FFFFh	Reserved	Reserved

<modify section 5.26 to change SFSC to SPC-4>

5.26 Security Send command – NVM Command Set Specific

The Security Send command is used to transfer security protocol data to the controller. The data structure transferred to the controller as part of this command contains security protocol specific commands to be performed by the controller. The data structure transferred may also contain data or parameters associated with the security protocol commands. Status and data that is to be returned to the host for the security protocol commands submitted by a Security Send command are retrieved with the Security Receive command defined in section 5.25.

The association between a Security Send command and subsequent Security Receive command is Security Protocol field dependent as defined in SPC-4 ~~SFSC~~.

The fields used are Data Pointer, Command Dword 10, and Command Dword 11 fields. All other command specific fields are reserved.

Figure 337: Security Send – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 105 for the definition of this field.

Figure 338: Security Send – Command Dword 10

Bits	Description
31:24	**Security Protocol (SECP):** This field specifies the security protocol as defined in SPC-4 ~~SFSC~~. The controller shall abort the command with status Invalid Field in Command if a reserved value of the
23:16	**SP Specific 1 (SPSP1):** The value of this field contains bits 15:08 of the Security Protocol Specific field as defined in SPC-4 ~~SFSC~~.
15:08	**SP Specific 0 (SPSP0):** The value of this field contains bits 07:00 of the Security Protocol Specific field as defined in SPC-4 ~~SFSC~~.
07:00	**NVMe Security Specific Field (NSSF):** Refer to Figure 336 for definition of this field for Security Protocol EAh. For all other Security Protocols this field is reserved.

Figure 339: Security Send – Command Dword 11

Bits	Description
31:00	**Transfer Length (TL):** The value of this field is specific to the Security Protocol Out command with the INC_512 field cleared to 0h as defined in SPC-4 ~~SFSC where INC_512 = 0~~.

5.26.1 Command Completion

If the command is completed, then the controller shall post a completion queue entry to the Admin Completion Queue indicating the status for the command.

<Modify Figure 329: Format NVM – Command Specific Status Values>

Value	Description
0Ah	**Invalid Format:** The format specified is invalid. This may be due to various conditions, including: 1) specifying an invalid LBA Format number; 2) enabling protection information when there is not sufficient metadata per LBA; 3) the specified format is not available in the current configuration; or 4) invalid security state (refer to TCG Storage Interface Interactions S~~s~~pecification), etc.

<Modify Figure 347: NSID Types and Relationship to Namespace>

Valid NSID Type	NSID relationship to namespace	Reference
Unallocated	Does not refer to any namespace that exists in the NVM subsystem	6.1.3
Allocated	Refers to a namespace that exists in the NVM subsystem	6.1.3
Inactive	Does not refer to a namespace that is attached to the ~~this~~ controller[1]	6.1.4
Active	Refers to a namespace that is attached to this controller	6.1.4
NOTES: 1. If allocated, refers to a namespace that is not attached to the ~~this~~ controller. If unallocated, does not refer to any namespace.		

<Modify 6.7.1.1 Deallocate>

A logical block that has been deallocated using the Dataset Management command is no longer deallocated when the logical block is written. Read operations and Verify operations do not affect the deallocation status of a logical block. The value read from a deallocated logical block shall be deterministic; specifically, the value returned by subsequent reads of that logical block shall be the same until a write operation occurs to that logical block.

The values read from a deallocated logical block and its metadata (excluding protection information) shall be all bytes cleared to 0h (e.g., bits 2:0 in the DLFEAT field are set to 001b), all bytes set to FFh (e.g., bits 2:0 in the DLFEAT field are set to 010b), or the last data written to the associated logical block and its metadata, except that access is prohibited to all data and metadata values written before the most recent successful sanitize operation, if any. The Deallocate Logical Block Features (DLFEAT) field in the Identify Namespace data structure (refer to Figure 245) may report the values read from a deallocated logical block and its metadata.

The values read from a deallocated or unwritten logical block's protection information field shall:

- have the Guard field value set to FFFFh or set to the CRC for the value read from the deallocated logical block and its metadata (excluding protection information) (e.g., cleared to 0h if the value read is all bytes cleared to 0h); and
- have the Application Tag field value set to FFFFh and the Reference Tag field value set to FFFFFFFFh (indicating the protection information shall not be checked).

Using the Error Recovery feature (refer to section 5.21.1.5), host software may enable an error to be returned if a deallocated or unwritten logical block is read. If this error is supported for the namespace and enabled, then a Read, Verify, or Compare command that includes a deallocated or unwritten logical block shall abort ~~fail~~ with the Unwritten or Deallocated Logical Block status code. Note: Legacy software may not handle an error for this case.

Note: The operation of the Deallocate function is similar to the ATA DATA SET MANAGEMENT with Trim feature described in ACS-4 and SCSI UNMAP command described in SBC-3.

<Modify Figure 382: Reservation Register – Command Dword 10>

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:30</td><td>Change Persist Through Power Loss State (CPTPL): This field allows the Persist Through Power Loss (PTPL) state associated with the namespace to be modified as a side effect of processing this command. If ~~a saveable value is supported for~~ the Reservation Persistence Feature (refer to section 5.21.1.28) is saveable, then any change to the PTPL state as a result of processing this command shall be applied to both the current value and the saved ~~saveable~~ value of that feature.
<table>
<tr><th>CPTPL Value</th><th>Description</th></tr>
<tr><td>00b</td><td>No change to PTPL state</td></tr>
<tr><td>01b</td><td>Reserved</td></tr>
<tr><td>10b</td><td>Set PTPL state to ‘0’. Reservations are released and registrants are cleared on a power on.</td></tr>
<tr><td>11b</td><td>Set PTPL state to ‘1’. Reservations and registrants persist across a power loss.</td></tr>
</table></td></tr>
<tr><td>29:04</td><td>Reserved</td></tr>
<tr><td>03</td><td>Ignore Existing Key (IEKEY): If this bit is set to a ‘1’, then Reservation Register Action (RREGA) field values that use the Current Reservation Key (CRKEY) shall succeed regardless of the value of the Current Reservation Key field in the command (i.e., the current reservation key is not checked).</td></tr>
<tr><td>02:00</td><td>Reservation Register Action (RREGA): This field specifies the registration action that is performed by the command.
<table>
<tr><th>RREGA Value</th><th>Description</th></tr>
<tr><td>000b</td><td>Register Reservation Key</td></tr>
<tr><td>001b</td><td>Unregister Reservation Key</td></tr>
<tr><td>010b</td><td>Replace Reservation Key</td></tr>
<tr><td>011b to 111b</td><td>Reserved</td></tr>
</table></td></tr>
</table>

<Modify 6.16 Write Uncorrectable command>

The Write Uncorrectable command is used to mark a range of logical blocks as invalid. When the specified logical block(s) are read after this operation, a failure is returned with Unrecovered Read Error status. To clear the invalid logical block status, a write operation is performed on those logical blocks.

The fields used are Command Dword 10, Command Dword 11, and Command Dword 12 fields. All other command specific fields are reserved.

Figure 407: Write Uncorrectable – Command Dword 10 and Command Dword 11

Bits	Description
63:00	**Starting LBA (SLBA):** This field specifies the 64-bit address of the first logical block to become uncorrectable ~~be marked as invalid~~ as part of the operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63: 32.

Figure 408: Write Uncorrectable – Command Dword 12

Bits	Description
31:16	Reserved
15:00	**Number of Logical Blocks (NLB):** This field specifies the number of logical blocks to become uncorrectable ~~be marked as invalid~~. This is a 0's based value.

<Modify 7.8 Feature Values>

The Get Features command, (refer to section 5.13), and Set Features command, (refer to section 5.21), may be used to read and modify operating parameters of the controller. The operating parameters are grouped and identified by Feature Identifiers. Each Feature Identifier contains one or more attributes that may affect the behavior of the Feature.

If bit 4 is set to '1' in the Optional NVM Command Support (ONCS) field of the Identify Controller data structure in Figure 247, then for each Feature, there are three settings: default, saved ~~saveable~~, and current. If bit 4 is cleared to '0' in the Optional NVM Command Support field of the Identify Controller data structure, then the controller only supports a current and default value for each Feature. In this case, the current value may be persistent across power states based on the information specified in Figure 271 and Figure 272.

~~The default value for each Feature is vendor specific and set by the manufacturer unless otherwise specified. The default value is not changeable. The saveable value is the value that the Feature has after a power on or reset event. The controller may not support a saveable value for a Feature; this is discovered by using the 'supported capabilities' value in the Select field in Get Features. If the controller does not support a saveable value for a Feature, then the default value is used after a power on or reset event. The current value for a Feature is the value in active use by the controller for that Feature~~

~~A Set Features command may be used to modify the saveable value, if supported, and the current value for a Feature. A Get Features command may be used to read the default value, the saveable value, if supported, and the current value for a Feature. If the controller does not support a saveable value for a Feature, then the default value is returned for the saveable value in a Get Features command.~~

Each Feature has supported capabilities (refer to figure NEW), which are discovered by using the 'supported capabilities' value in the Select field in Get Features (refer to Figure 182).

The default value for each Feature is vendor specific and set by the manufacturer unless otherwise specified. The default value is not changeable.

The saved value is the value that the Feature has after a Controller Level Reset. A Feature may be saveable. If a Feature is not saveable, then:

a) the default value is used after a Controller Level Reset; and
b) a Get Features command to read the saved value returns the default value.

The current value for a Feature is the value in active use by the controller for that Feature.

A Set Feature command uses the value specified by the command to set:
a) The current value for that Feature; or
b) The current value for that Feature, and the saved value for that Feature, if that Feature is saveable.

Feature settings may apply to:

a) the controller (i.e., the feature is not namespace specific); or b) a namespace (i.e., the feature is namespace specific).

For feature values that apply to the controller:

a) if the NSID field is cleared to 0h or set to FFFFFFFFh, then:

- the Set Features command shall set the specified feature value for the controller; and
- the Get Features command shall return the current setting of the requested feature value for the controller; and

b) if the NSID field is set to a valid namespace identifier (refer to section 6.1), then:

- the Set Features command shall abort ~~fail~~ with a status code of Feature Not Namespace Specific; and
- the Get Features command shall return the current setting of the requested feature value for the controller.

For feature values that apply to a namespace:

a) if the NSID field is set to an active namespace identifier (refer to section 6.1), then:

- the Set Features command shall set the specified feature value of the specified namespace; and
- the Get Features command shall return the current setting of the requested feature value for the specified namespace;

b) if the NSID field is set to FFFFFFFFh, then:

- the Set Features command shall, unless otherwise specified, set the specified feature value for all namespaces attached to the controller processing the command; and
- the Get Features command shall, unless otherwise specified in section 5.21.1, fail with a status code of Invalid Namespace or Format;

and

c) if the NSID field is set to any other value, then the Set Features command and the Get Features command shall abort ~~fail~~ as described in the rules for namespace identifier usage in Figure 105.

…

<Modify 7.14 Privileged Actions >

Privileged actions are actions (e.g., command, register write) that affect or have the potential to affect the state of the entire NVM subsystem and not only the controller and/or namespace with which the action is associated.

Admin commands that are privileged include Namespace Management, Namespace Attachment, Virtualization Management, Format NVM, Set Features with Feature Identifier 17h (i.e., Sanitize Config, refer to section 5.21.1.23), and Sanitize. A privileged register action is NVM Subsystem Reset ~~subsystem reset~~. Vendor specific commands and registers may also be privileged.

<Modify 8.3.1.5 Control of Protection Information Checking - PRCHK>

< ….. change to bullet list for bits 0,1,2; combine bit 0 requirements under 1 bullet for types 0/1/2>

< …… see also related changes in section 6.17….>

8.3.1.5 Control of Protection Information Checking - PRCHK

Checking of protection information consists of the following operations performed by the controller.

- If bit 2 of the Protection Information Check (PRCHK) field of the command is set to ‘1’, then the controller compares the protection information Guard field to the CRC-16 computed over the logical block data.
- If bit 1 of the PRCHK field is set to ‘1’, then the controller compares unmasked bits in the protection

information Application Tag field to the Logical Block Application Tag (LBAT) field in the command. A bit in the protection information Application Tag field is masked if the corresponding bit is cleared to '0' in the Logical Block Application Tag Mask (LBATM) field or the Expected Logical Block Application Tag Mask (ELBATM) field of the command.

- If bit 0 of the PRCHK field is set to '1', then the controller compares the protection Information Reference Tag field to the computed reference tag. The computed reference tag depends on the Protection Type:
  - If the namespace is formatted for Type 1 protection, the value of the computed reference tag for the first logical block of the command is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command, and the computed reference tag is incremented for each subsequent logical block. The controller shall complete the command with a status of Invalid Protection Information if the ILBRT field or the EILBRT field does not match the least significant four bytes of the SLBA field.

    Note: Unlike SCSI Protection Information Type 1 protection which implicitly uses the least significant four bytes of the LBA, the controller always uses the ILBRT or EILBRT field and requires host software to initialize the ILBRT or EILBRT field to the least significant four bytes of the LBA when Type 1 protection is used.
  - If the namespace is formatted for Type 2 protection, the value of the computed reference tag for the first logical block of the command is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command, and the computed reference tag is incremented for each subsequent logical block.
  - If the namespace is formatted for Type 3 protection, the value of the computed reference tag for the first LBA of the command and all subsequent logical blocks is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command.

~~For Type 1 protection, if bit 0 of the PRCHK field is set to '1', then the controller compares the protection information Reference Tag field to the computed reference tag. The value of the computed reference tag for the first LBA of the command is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command. If the namespace is formatted for Type 1 or Type 2 protection, the computed reference tag is incremented for each subsequent logical block. If the namespace is formatted for Type 3 protection, the reference tag for each subsequent logic block remains the same as the initial reference tag. Unlike SCSI Protection Information Type 1 protection which implicitly uses the least significant four bytes of the LBA, the controller always uses the ILBRT or EILBRT field and requires host software to initialize the ILBRT or EILBRT field to the least significant four bytes of the LBA when Type 1 protection is used. In Type 1 protection, the controller should check the ILBRT field or the EILBRT field for the correct value; if the value does not match the least significant four bytes of the LBA, then the controller completes the command with a status of Invalid Protection Information.~~

~~For Type 2 protection, if bit 0 of the PRCHK field is set to '1', then the controller compares the protection information Reference Tag field from each logical block to the computed reference tag.~~

~~For Type 3 protection, if bit 0 of the PRCHK field is set to '1', then the command should be aborted with status Invalid Protection Information, but may be aborted with status Invalid Field in Command.~~

Protection checking may be disabled as a side effect of the value of the protection information Application Tag and Reference Tag fields regardless of the state of the PRCHK field in the command. If the namespace is formatted for Type 1 or Type 2 protection, then all protection information checks are disabled regardless of the state of the PRCHK field when the protection information Application Tag has a value of FFFFh. If the namespace is formatted for Type 3 protection, then all protection information checks are disabled regardless of the state of the PRCHK field when the protection information Application Tag has a value of FFFFh and the protection information Reference Tag has a value of FFFFFFFFh.

Inserted protection information consists of the computed CRC-16 in the Guard field, the LBAT field value in the Application Tag, and the computed reference tag in the Reference Tag field.

<Modify 8.4 Power Management>

…

The host may dynamically modify the power state using the Set Features command and determine the current power state using the Get Features command. The host may directly transition between any two supported power states. The Entry Latency (ENTLAT) field in the Power State Descriptor data structure ~~power management descriptor~~ indicates the maximum amount of time in microseconds to enter that power state and the Exit Latency (EXLAT) field indicates the maximum amount of time in microseconds to exit that state.

…

<Modify 8.5 Virtualization Enhancements (Optional)>

…

Primary and secondary controllers may implement all features of this specification, except where commands are defined as being only supported by a primary controller ~~clearly marked as primary controller only~~. It is recommended that only primary controllers support the privileged actions described in section 7.14 so that untrusted hosts using secondary controllers do not impact the entire NVM subsystem state.

…

<Modify 8.17 Endurance Groups (Optional)>

Endurance may be managed within a single NVM Set (refer to section 4.9) or across a collection of NVM Sets. Each NVM Set is associated with an Endurance Group (refer to Figure 250). If two or more NVM Sets have the same Endurance Group Identifier, then endurance is managed by the NVM subsystem across that collection of NVM Sets. If only one NVM Set is associated with a specific Endurance Group Identifier, then endurance is managed locally to that NVM Set.

An Endurance Group Identifier is a 16-bit value that specifies the Endurance Group with which an action is associated. An Endurance Group Identifier value of 0h is reserved and is not a valid Endurance Group Identifier. Unless otherwise specified, if the host specifies an Endurance Group Identifier cleared to 0h for a command that requires an Endurance Group Identifier, then that command shall abort ~~fail~~ with a status code of Invalid Field in Command.

The endurance information for an Endurance Group is specified in the Endurance Group Information log page (refer to section 5.14.1.9).

…

<Modify 8.19 Namespace Write Protection (Optional)>

…

The results of using Namespace Write Protection in combination with an external write protection system (e.g., TCG Storage Interface Interactions Specification ~~specification~~) are outside the scope of this

specification.

…

<Modify a portion of Figure 494: ANA effects on Command Processing>

Command	ANA State	Effects on command processing
…		
Get Log Page	ANA Inaccessible, ANA Persistent Loss, or ANA Change	<remove bullet list> The following log pages are affected: Error Information (i.e., 01h): The log page ~~may~~ is not required to contain entries ~~only~~ for namespaces whose relationship to the controller processing the command is in the ~~-~~ANA Inaccessible state (refer to section 8.20.3.3), the ANA Persistent Loss state (refer to section 8.20.3.4), or the ANA Change state (refer to section 8.20.3.5). ~~ANA Optimized state (refer to section 8.20.3.1) or the ANA Non-optimized state (refer to section 8.20.3.2).~~
…		
Set Features	ANA Inaccessible	The saving of features shall not be supported and the following feature identifiers are not available[1]: a) LBA Range Type (i.e., 03h); b) Error Recovery (i.e., 05h); c) Write Atomicity Normal (i.e., 0Ah); d) Reservation Notification Mask (i.e., 82h); and e) Reservation Persistence (i.e., 83h). If the NSID is set to FFFFFFFFh, then the command shall abort ~~fail~~ with a status code of Asymmetric Access Inaccessible (refer to section 8.20.3.3).
	ANA Change	The saving of features shall not be supported and the following feature identifiers are not available[1]: a) LBA Range Type (i.e., 03h); b) Error Recovery (i.e., 05h); c) Write Atomicity Normal (i.e., 0Ah); d) Reservation Notification Mask (i.e., 82h); and e) Reservation Persistence (i.e., 83h). If the NSID is set to FFFFFFFFh, then the command shall abort ~~fail~~ with a status code of Asymmetric Access Transition (refer to section 8.20.3.5).
	ANA Persistent Loss	This command shall abort ~~fail~~ with a status code of Asymmetric Access Persistent Loss (refer to section 8.20.3.4).
NOTES: 1. If the ANA state is ANA Inaccessible State, then commands that use feature identifiers that are not available shall abort ~~fail~~ with a status code of Asymmetric Access Inaccessible. If the ANA state is ANA Persistent Loss State, then commands that use feature identifiers that are not available shall abort ~~fail~~ with a status code of Asymmetric Access Persistent Loss. If the ANA state is ANA Change State, then commands that use feature identifiers that are not available shall abort ~~fail~~ with a status code of Asymmetric Access Transition.		

<Modify 8.22 Get LBA Status (Optional)>

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 is able to 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 Information 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 Information Attributes Feature (refer to section 5.21.1.21) 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 287).

…