LEGAL NOTICE:

This erratum is proprietary to the NVM Express, Inc. (also referred to as “Company”) and/or its successors and assigns.

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

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **“© 2008 to 2024 NVM Express, Inc. ALL RIGHTS RESERVED.”** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

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

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

The NVM Express® design mark is a registered trademark of NVM Express, Inc.
PCI-SIG®, PCI Express®, and PCIe® are registered trademarks of PCI-SIG.
InfiniBand™ is a trademark and servicemark of the InfiniBand Trade Association.

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

NVM Express® Technical Errata

Errata ID	**122**
Revision Date	**07/28/2024**
Affected Spec Ver.	**NVM Express® Base Specification Revision 2.0d** **NVM Express® NVM Command Set Specification Revision 1.0d** **NVM Express® Management Interface Specification Revision 1.2d** **NVM Express Boot Specification 1.0**
Corrected Spec Ver.	
Referenced Specifications	**TP4034a Dispersed Namespaces**

Errata Author(s)

Name	Company
Mike Allison, Judy Brock, Bill Martin,	Samsung
David Black, Austin Bolen, Doug Farley	Dell
Dan Hubbard	Micron
Paul Suhler, Fred Knight	Kioxia
Andres Baez	Solidigm
Phil Cayton	Intel

Errata Overview

This ECN updates and clarifies various text within the NVM Express Base Specification Revision 2.0d, NVM Express NVM Command Set Specification Revision 1.0d, NVM Express Boot Specification 1.0, and the NVM Express Management Interface Specification 1.2d.

Revision History

Revision Date	Change Description
01/31/2024	Initial creation with the solutions for: • Bug 291 SGL Identifiers in wrong bit order • Bug 292 Smallest Format Progress Indicator forgets bit 7 value
02/06/2024	Added the solution for: • Bug 290 DHCP Root-Path Override Flags bit misspelled SGL Identifiers in wrong bit order
02/07/2024	Added the solution for: • Bug 293 Namespace Write Protection Authentication field is undefined
02/14/2024	Added the solution for: • Bug 300 Resolve Miscellaneous Misspellings and Duplicate Words • Bug 294 NOWS Field Description is Ambiguous
02/21/2024	Added the solution for: • Bug 299 Version 2.0d Figure 203 Get Log Page - Log Page Identifiers lost the footnotes • Bug 301 Firmware Update overlap detection text implies it's both mandatory and optional • Due to additional changes the reintegration of Bug 294 NOWS Field Description is Ambiguous
02/28/2024	Added the solution for: • Bug 296 Get LBA Status clarification
03/06/2024	Added the solution for: • Bug 306 higher level should be hyphenated • Bug 18 TP6025a introduced issue in DLEN
03/16/2024	Added the solution for: • Bug 305 Command Scope Clarifications
3/20/2024	Added the solution for: • Bug 310 Controller Properties - Property Definition table limits the size of Transport Specific range
4/18/2024	Added the solution for: • Bug 319 Format NVM may deallocate or allocate logical blocks • Bug 321 Changeable Features Clarifications
5/1/2024	Added the solution for: • Bug 311 Invalid Opcode should be Invalid Command Opcode
5/8/2024	Updated the bug solution 18 with new changes.
5/10/2024	Updated section 6.2 of the NVMe-MI specification. Aligned the blue coloring to be consistent.
5/15/2024	Added the solution for: • Bug 308 HFI Primary and Secondary Indices utilization is unclear • Bug 330 Reservation Report Corrections • Bug 124 Reservation text has some errors
5/16/2024	Added in TP4034a text for the reservation changes. Editorial changes made in Review in the NVM Express Technical WG.
5/21/2024	Split up Figure 35 (Property Definitions).
5/22/2024	Corrected the Reservation text about NVM Express Base Specification, Revision 2.0 and earlier.
5/30/2024	Editorial changes during review.
7/7/2024	Integrated into published documents
7/15/2024	Correct coloring of NVMe-MI Figure 118 from Orange to Purple (text was not from a different ECN, it had been moved from one part of the sentence to another part – for clarity).
7/28/2024	Editorial changes during integration

Description of Changes

NVM Express Base Specification 2.0d:

Incompatible Changes:

- Changed Reservation Report command to be registrant-oriented rather than controller-oriented. E.G., changed all occurrences of “Registered Controller” to “Registrant”. This allows the command to return entries for all registrants of a namespace, including those that are not associated with any controller in the NVM subsystem (previously such registrants were not able to be reported). This is an incompatible change.
- The following data structure name changes and field name change occurred:
    - The “Number of Registered Controllers (REGCTL)” field name was renamed to “Number of Registrants (REGSTRNT)”
    - The “Registered Controller” data structure name was renamed to “Registrant” data structure
    - The “Registered Controller Extended” data structure name was changed to “Registrant Extended” data structure
- Expanded meaning of value FFFFh in CNTLID field to be used in the case of any registrant whose host identifier is not associated with a controller in the NVM subsystem. This now covers PCIe controllers, NVMe-oF static controllers and NVMe-OF dynamic controllers alike. This may be considered an incompatible change.

Editorial Changes:

- Corrected the bit ordering of the SGL descriptors SGL Identifiers.
- Corrected the bits the Smallest Format Progress Indicator field uses in the FPI field.
- Corrected the misnamed use of the Write Protection Control field.
- Resolved several misspelled or duplicate words
- Added the footnotes to Figure 203: Get Log Page – Log Page Identifiers, which were accidently removed between 2.0c and 2.0d.
- Clarified that Firmware Update overlap checking is optional.
- Hyphenated the phrase higher-level software.
- Clarified the setting multiple bits in the Command Scope field in the Commands Supported and Effects log page.
- Corrected the Vendor Specific offset and clarified the expected value for the empty field definitions.
- Added text to Get Features command to clarify that a feature is considered to be Changeable (i.e. reports support for the Changeable Capability in Get Features command when the SEL filed is set to 011b) if it is capable of supporting a changeable value even though it may at times be set to a value that is unchangeable (e.g., Namespace Write Protection Config, Boot Partition Write Protection Config features are Changeable but may be set to unchangeable values).
- Clarify that for Preempt and Abort, that it is the reservation that is preempted and that outstanding commands are what are aborted (i.e., there is no such thing as aborting a reservation).

NVM Express NVM Command Set Specification 1.0d:

Editorial Changes:

- Clarified the definition of the NOWs field and the NPWG field to address that the NOWS field is in logical block units and the MDTS field is in memory page size with different definitions as to how many logical blocks are transferred.

- Clarified that it is the discovery of untracked LBAs by the controller that may or may not cause those LBAs to be added to the tracked list (as opposed to after the host discovers them (as a result of a Get LBA Status command returning them).
- Added Format NVM to the lists of commands that may deallocate or may allocate logical blocks.

NVM Express Boot Specification 1.0:

Editorial Changes:

- Fixed the name in the text when referencing the SNSS DHCP Root-Path Override Flag bit.
- Added a section to Appendix A for HFI handling and reference from HFI Descriptor.

NVM Express Management Interface Specification 1.2d:

Incompatible Changes:

- In the Management Interface Command Set, fixed an issue where the DOFST/DLEN checking logic specified an error if DOFST plus LEN is “greater than or equal to” the size of the Management Endpoint Buffer but should have said “greater than”.

Editorial Changes:

- Clarified that in the NVM Express Admin Command Set that Management Endpoints do not return an error if the DLEN field is 0h for commands that indicate the transfer data in the Data Transfer subfield for the opcode field of the NVMe Admin Command but don’t actually have transfer data (e.g., certain Get Features/Set Features command features.
- For the VPD Read command and VPD Write command the Management Interface Command Set, recommended that the case where the DOFST field is set to 4096 and the DLEN filed is cleared to 0h result in an Invalid Parameter Error Response being returned.
- Clarified that if multiple Response Message Status values apply due to multiple unique error conditions being present or due to multiple Response Message Status values applying for a single error condition, then the Responder chooses one of those applicable Response Message Status values to return.
- For Invalid Command Size, clarified that Request Data being present when not expected does not count as an error due to too much Request Data.
- Fixed the Request Message Response status of “Invalid Opcode” to be “Invalid Command Opcode”.
- Removed this use of “it” with the actual noun for clarity.

Note:

BLACK text indicates unchanged text. **BLUE** text indicates newly inserted text. ~~**RED stricken**~~ text indicates deleted text; **ORANGE** text indicates changes from another ECN. **Purple** text indicates destination of moved text without changes. ~~**Purple stricken**~~ text indicates source of moved text without changes. **GREEN** text indicates editor notes.

Description of NVM Express Base Specification 2.0d changes

<Change all occurrences of "higher level software" to "higher-level software">

Modify section 3 as shown below:

3 Admin Command Set

3.1 NVM Controller Architecture

…

3.1.3 Controller Properties

…

Figure 35: Property Definition

Offset (OFST)	Size (in bytes)	I/O Controller[1]	Admin. Controller[1]	Discovery Controller[1]	Name
0h	8	M	M	M	**CAP:** Controller Capabilities
…					
E14h	4	O[3]	O[3]	R	**PMRMSCL:** Persistent Memory Region Controller Memory Space Control Lower
E18h	4	O[3]	O[3]	R	**PMRMSCU:** Persistent Memory Region Controller Memory Space Control Upper
E1Ch		R	R	R	Reserved
1000h		Transport Specific: • Refer to Figure TBD1 for Memory-Based transport implementations. • Refer to Figure TBD2 for Message-Based transport implementations.			
~~1000h~~		~~T~~	~~T~~	~~T~~	~~Transport Specific~~
~~1300H~~		~~O~~	~~O~~	~~O~~	~~Vendor Specific~~

Notes:

1. O/M/~~P~~R definition: O = Optional, M = Mandatory, R = Reserved~~, T = Transport Specific~~
2. Mandatory for memory-based transport implementations. Reserved for message-based transport implementations.
3. Optional for memory-based transport implementations. Reserved for message-based transport implementations.
4. Determined by the transport (e.g., the offset calculation formula Offset (1000h + ((2y) * (4 << CAP.DSTRD))) for the memory-based PCIe transport).

Figure TBD1: Memory-Based Property Definition

Offset (OFST)	Size (in bytes)	I/O Controller[1]	Admin. Controller[1]	Discovery Controller[1]	Name
1000h	Variable[2]	T	T	T	Transport Specific (e.g., PCIe doorbell registers as specified in the NVMe over PCIe Transport Specification)

Figure TBD1: Memory-Based Property Definition

Offset (OFST)	Size (in bytes)	I/O Controller[1]	Admin. Controller[1]	Discovery Controller[1]	Name
1000h + Variable[2]		O	O	O	Vendor Specific

Notes:

1. O/T definition: O = Optional, T = Transport Specific
2. Determined by the transport (e.g., the offset calculation formula Offset (1000h + ((2y) * (4 << CAP.DSTRD))) for the PCIe transport)

Figure TBD2: Message-Based Property Definition

Offset (OFST)	Size (in bytes)	I/O Controller[1]	Admin. Controller[1]	Discovery Controller[1]	Name
1000h	300h	R	R	R	Reserved for Fabrics
1300h		O	O	O	Vendor Specific

Notes:

1. O/R definition: O = Optional,-R = Reserved

…

3.9 Keep Alive

…

3.9.4 Traffic Based Keep Alive

…

3.9.4.2 Traffic Based Keep Alive on the Host

The host may use Traffic Based Keep Alive only if the controller is using Traffic Based Keep Alive. The host should check for a command completion queue entry for any Admin commands and I/O commands at half of the time equal to the Keep Alive Timeout value to account for ~~for~~ delays (e.g., transport roundtrip times, transport delays, command processing times, and the Keep Alive Timer granularity). To prevent the controller from detecting a Keep Alive Timeout during the use of Traffic Based Keep Alive on the host, if no Admin command and no I/O command is sent to the controller during half of the Keep Alive Timeout Interval, the host should send a Keep Alive command.

…

3.11 Firmware Update Process

…

The process for a firmware update to be activated on a domain without a reset is:

1. The host issues a Firmware Image Download command to download the firmware image to a controller. There may be multiple portions of the firmware image to download, thus the offset for each portion of the firmware image being downloaded on that controller is specified in the Firmware Image Download command. The data provided in the Firmware Image Download command should conform to the Firmware Update Granularity indicated in the Identify Controller data structure or the firmware update may fail;

   a. …

2. The controller completes the Firmware Commit command. The following actions are taken in

certain error scenarios:

a. If the firmware image is invalid, then the controller aborts the command with an appropriate status code (e.g., Invalid Firmware Image);
b. …
c. If the firmware activation was not successful because an NVM Subsystem Reset is required to activate this firmware image, then the controller aborts the command with a status code of Firmware Activation Requires NVM Subsystem Reset and the firmware ~~firrmware~~ image is applied at the next NVM Subsystem Reset;
d. …

…

Modify section 4 as shown below:

4 Data Structures

…

4.1 Data Layout

…

4.1.2 Scatter Gather List (SGL)

…

Figure 115: Generic SGL Descriptor Format

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">…</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td>SGL Descriptor Type (refer to Figure 116)</td></tr>
<tr><td>03:00</td><td>SGL Descriptor Sub Type (refer to Figure 117)</td></tr>
<tr><td>~~07:04~~</td><td>~~SGL Descriptor Type (refer to Figure 116)~~</td></tr>
</table></td></tr>
</table>

…

Figure 118: SGL Data Block descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">…</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 0h as specified in Figure 116.</td></tr>
<tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 117.</td></tr>
<tr><td>~~07:04~~</td><td>~~<b>SGL Descriptor Type:</b> 0h as specified in Figure 116.~~</td></tr>
</table></td></tr>
</table>

…

Figure 119: SGL Bit Bucket descriptor

Bytes	Description
…	

Figure 119: SGL Bit Bucket descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 1h as specified in Figure 116.</td></tr>
<tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 117.</td></tr>
<tr><td>~~07:04~~</td><td>~~<b>SGL Descriptor Type:</b> 1h as specified in Figure 116.~~</td></tr>
</table></td></tr>
</table>

…

Figure 120: SGL Segment descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">…</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 2h as specified in Figure 116.</td></tr>
<tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 117.</td></tr>
<tr><td>~~07:04~~</td><td>~~<b>SGL Descriptor Type:</b> 2h as specified in Figure 116.~~</td></tr>
</table></td></tr>
</table>

…

Figure 121: SGL Last Segment descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">…</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 3h as specified in Figure 116.</td></tr>
<tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 117.</td></tr>
<tr><td>~~07:04~~</td><td>~~<b>SGL Descriptor Type:</b> 3h as specified in Figure 116.~~</td></tr>
</table></td></tr>
</table>

…

Figure 122: Keyed SGL Data Block descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">…</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 4h as specified in Figure 116.</td></tr>
<tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 117.</td></tr>
<tr><td>~~07:04~~</td><td>~~<b>SGL Descriptor Type:</b> 4h as specified in Figure 116.~~</td></tr>
</table></td></tr>
</table>

…

Figure 123: Transport SGL Data Block descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2">…</td></tr>
<tr><td>15</td><td><b>SGL Identifier:</b> The definition of this field is described in the table below.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>07:04</td><td><b>SGL Descriptor Type:</b> 5h as specified in Figure 116.</td></tr>
<tr><td>03:00</td><td><b>SGL Descriptor Sub Type:</b> Valid values are specified in Figure 117.</td></tr>
<tr><td>~~07:04~~</td><td>~~<b>SGL Descriptor Type:</b> 5h as specified in Figure 116.~~</td></tr>
</table></td></tr>
</table>

…

Modify section 5 as shown below:

5 Admin Command Set

…

Figure 140: Opcodes for Admin Commands

Opcode by Field			Combined Opcode [1]	Namespace Identifier Used [2]	Command	Command Set Specific [8]
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[3]				
…						
Notes: 1. Opcodes not listed are reserved. … 10. Use of the Namespace Identifier field is specified further ~~fruther~~ in section 5.16.1 and Figure 203.						

…

5.12 Firmware Commit command

…

Figure 184: Firmware Commit – Command Specific Status Values

Value	Description
…	
14h	**Overlapping Range**: This error is indicated if the controller detects that the firmware image has overlapping ranges.
…	

…

5.13 Firmware Image Download command

…

Figure 188: Firmware Image Download – Command Specific Status Values

Value	Description
…	
14h	**Overlapping Range**: This error is indicated if the controller detects that the firmware image has overlapping ranges. This error may indicate that the granularity or alignment of the firmware image downloaded does not conform to the Firmware Update Granularity field indicated in the Identify Controller data structure.
…	

…

5.15 Get Features command

…

5.15.2 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 011b, then Dword 0 of the completion queue entry may contain feature-dependent information (refer to section 5.27.1).

If the Select field is set to 011b, then Figure 196 describes the contents of Dword 0 of the completion queue entry.

Figure 196: Completion Queue Entry Dword 0 when Select is set to 011b

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. Changing a feature value is described in section 5.27
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.

If the controller supports any changeable value of any attribute of a feature, then the controller reports that feature as changeable (i.e., the controller sets the Changeable bit to '1' in Completion Queue Entry Dword 0 for a Get Features command that specifies that feature and has the Select field set to 011b), even if the feature has been set to a value or values that are not changeable.

For some features (e.g., Namespace Write Protection Config (refer to section 5.27.1.28), Boot Partition Write Protection Config), <note to editor: do not include ", Boot Partition Write Protection Config" text snippet immediately above in the ECN. Only include it in .NEXT file> the changeability of feature values is value-dependent (e.g., the Permanent Write Protect (i.e., 011b) value of the Write Protection State in the Namespace Write Protection Config feature is not changeable).

…

5.16 Get Log Page command

…

Figure 203: Get Log Page – Log Page Identifiers

Log Page Identifier	CSI[7]	Scope	Log Page Name	Reference Section
00h	Y	Controller	Supported Log Pages	5.16.1.1
01h	N	Controller	Error Information	5.16.1.2
02h	N	Controller[1] Namespace[2]	SMART / Health Information	5.16.1.3
03h	N	Domain / NVM subsystem[6]	Firmware Slot Information	5.16.1.4
04h	N	Controller	Changed Namespace List	5.16.1.5
05h	Y	Controller	Commands Supported and Effects	5.16.1.6
06h	N	Controller[3] Domain / NVM subsystem[4, 6]	Device Self-test[5]	5.16.1.7
07h	N	Vendor Specific	Telemetry Host-Initiated[5]	5.16.1.8
08h	N	Vendor Specific	Telemetry Controller-Initiated[5]	5.16.1.9
09h	N	Domain / NVM subsystem[6]	Endurance Group Information	5.16.1.10

Figure 203: Get Log Page – Log Page Identifiers

Log Page Identifier	CSI[7]	Scope	Log Page Name	Reference Section
0Ah	N	Domain / NVM subsystem [6]	Predictable Latency Per NVM Set	5.16.1.11
0Bh	N	Domain / NVM subsystem [6]	Predictable Latency Event Aggregate	5.16.1.12
0Ch	N	Controller	Asymmetric Namespace Access	5.16.1.13
0Dh	N	NVM subsystem	Persistent Event Log [5]	5.16.1.14
0Eh	Refer to the NVM Command Set			
0Fh	N	Domain / NVM subsystem [6]	Endurance Group Event Aggregate	5.16.1.15
10h	N	Domain / NVM subsystem [5, 6]	Media Unit Status	5.16.1.16
11h	N	Domain / NVM subsystem [6]	Supported Capacity Configuration List	5.16.1.17
12h	Y	Controller	Feature Identifiers Supported and Effects	5.16.1.18
13h	N	Controller	NVMe-MI Commands Supported and Effects	5.16.1.19
14h	Y	NVM subsystem	Command and Feature Lockdown [5]	5.16.1.20
15h	N	Controller	Boot Partition	5.16.1.21
16h	N	Endurance Group	Rotational Media Information	5.16.1.22
17h to 6Fh	Reserved			
70h	N	Controller	Discovery	5.16.1.23
71h to 7Fh	Reserved			
80h	N	Controller	Reservation Notification	5.16.1.24
81h	N	NVM subsystem	Sanitize Status	5.16.1.25
82h to BEh	I/O Command Set Specific			
BFh	Refer to the Zoned Namespace Command Set			
C0h to FFh	Vendor specific [5]			

Key:
Namespace = The log page contains information about a specific namespace.
Endurance Group = The log page contains information about a specific Endurance Group.
Controller = The log page contains information about the controller that is processing the command.
Domain = The log page contains information about the Domain.
NVM subsystem = The log page contains information about the NVM subsystem.
Vendor Specific = The log page contains information that is vendor specific.

Notes:
1. For namespace identifiers of 0h or FFFFFFFFh.
2. For namespace identifiers other than 0h or FFFFFFFFh.
3. Bit 0 is cleared to '0' in the DSTO field in the Identify Controller data structure (refer to Figure 276).
4. Bit 0 is set to '1' in the DSTO field in the Identify Controller data structure.
5. Selection of a UUID may be supported. Refer to section 8.25.
6. For NVM subsystems that support multiple domains (refer to the MDS bit in the Identify Controller data structure, Figure 276), Domain scope information is returned.
7. If multiple I/O Command Sets are supported, then the CSI field is used by the log page: Y = Yes, N = No. If Yes, then refer to the definition of the log page for details on usage.

…

5.16.1.6 Commands Supported and Effects (Log Page Identifier 05h)

Figure 212: Commands Supported and Effects Data Structure

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:20</td><td><b>Command Scope (CSP):</b> This field defines the scope for the associated command. If the value of this field is 0h then no scope is reported. A command may have multiple scopes depending on the effects of the command based on the parameters passed to that command. For a command that supports multiple scopes, multiple bits may set to ‘1’.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>11:6</td><td>Reserved</td></tr>
<tr><td>5</td><td><b>NVM Subsystem Scope</b>: If set to ‘1’, then the command performs actions that may impact the whole NVM subsystem. If cleared to ‘0’ and the CSP field is non-zero, then the command performs actions that does not impact the whole NVM subsystem.</td></tr>
<tr><td>4</td><td><b>Domain Scope</b>: If set to ‘1’, then the command performs actions that may impact a single Domain. If cleared to ‘0’ and the CSP field is non-zero, then the command performs actions that does not impact a single Domain.</td></tr>
<tr><td>3</td><td><b>Endurance Group Scope</b>: If set to ‘1’, then the command performs actions that may impact Endurance Groups. If cleared to ‘0’ and the CSP field is non-zero, then the command performs actions that does not impact a single Endurance Group.</td></tr>
<tr><td>2</td><td><b>NVM Set Scope</b>: If set to ‘1’, then the command performs actions that may impact NVM Sets. If cleared to ‘0’ and the CSP field is non-zero, then the command performs actions that do not impact NVM Sets.</td></tr>
<tr><td>1</td><td><b>Controller Scope</b>: If set to ‘1’, then the command performs actions that may impact controllers. If cleared to ‘0’ and the CSP field is non-zero, then the command performs actions that do not impact controllers.</td></tr>
<tr><td>0</td><td><b>Namespace Scope</b>: If set to ‘1’, then the command performs actions that may impact namespaces. If cleared to ‘0’ and the CSP field is non-zero, then the command performs actions that do not impact namespaces.</td></tr>
</table></td></tr>
<tr><td colspan="2">…</td></tr>
</table>

…

5.16.1.14 Persistent Event (Log Page Identifier 0Dh)

…

5.16.1.14.1 Persistent Event Log Events

…

5.16.1.14.1.8 Format NVM Completion Event (Event Type 08h)

…

Figure 268: Format NVM Completion Event Data Format (Event Type 08h)

Bytes	Description
03:00	**Namespace Identifier:** Contains the namespace identifier specified in the Format NVM command.
04	**Smallest Format Progress Indicator:** For a Format NVM command that formats a single namespace, this field contains the smallest ~~lowest~~ numerical value that was available for reporting in bits 6:0 of the FPI field ~~of~~ in the Identify Namespace data structure (i.e., if the format did not complete successfully and the FPI field is supported, then this field contains the percentage of the namespace that remained to be formatted at the time the Format NVM command completed, refer to Figure 344) during the format operation. For a Format NVM command that formats all namespaces this field shall be cleared to 0h.
…	

…

5.17 Identify command

…

5.17.2 Identify Data Structures

5.17.2.1 Identify Controller Data Structure (CNS 01h)

…

Figure 276: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O[1]	Admin[1]	Disc[1]	Description
…				
531	M	M	R	**Namespace Write Protection Capabilities (NWPC):** This field indicates the optional namespace write protection capabilities supported by the controller. Refer to section 8.12. Bits 7:3 are reserved. Bit 2 if set to ‘1’, then the controller supports the Permanent Write Protect state. If cleared to ‘0’, then the controller does not support the Permanent Write Protect state. If this bit is set to ‘1’, then the controller supports the ~~Namespace~~ Write Protection ~~Authentication~~ Control field (refer to section ~~8.18~~ 8.12.1). Bit 1 if set to ‘1’, then the controller supports the Write Protect Until Power Cycle state. If cleared to ‘0’, then the controller does not support Write Protect Until Power Cycle state. If this bit is set to ‘1’, then the controller supports the ~~Namespace~~ Write Protection ~~Authentication~~ Control field (refer to section ~~8.18~~ 8.12.1). Bit 0 if set to ‘1’, then the controller supports the No Write Protect and Write Protect namespace write protection states and may support the Write Protect Until Power Cycle state and Permanent Write Protect namespace write protection states (refer to section 8.12). If cleared to ‘0’, then the controller does not support Namespace Write Protection and bits 2:1 shall be cleared to 00b.
…				
Notes: 1. O/M/R definition: O = Optional, M = Mandatory, R = Reserved. 2. Mandatory for I/O controllers using a message-based transport. Reserved for controllers using a memory-based transport. 3. Mandatory for Discovery controllers that support explicit persistent connections. Reserved for Discovery controllers that do not support explicit persistent connections. 4. TBKAS bit is optional for Discovery controllers, and all other bits are reserved for Discovery controllers.				

…

5.27 Set Features command

…

If the controller does not support a changeable value for a Feature (~~e.g~~.i.e., the Feature is not changeable, refer to section 5.15.2), and a Set Features 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.

Refer to each feature description in section 5.27.1 for any additional requirements associated with that feature.

…

5.27.1 Feature Specific Information

…

5.27.1.28 Namespace Write Protection Config (Feature Identifier 84h)

…

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 controller shall abort the command with a status code of Feature Not Changeable.

…

Modify section 7 as shown below:

7 I/O Commands

…

7.2 Reservation Acquire command

The Reservation Acquire command is used to:

- acquire a reservation on a namespace~~,~~;
- preempt a reservation held on a namespace~~,~~; or
- preempt a reservation held on a namespace and abort ~~a reservation held on a~~ outstanding commands for that namespace.

The command uses Command Dword 10 and a Reservation Acquire data structure in memory. If the command uses PRPs for the data transfer, then PRP Entry 1 and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the SGL Entry 1 field is used. All other command specific fields are reserved.

…

7.5 Reservation Report command

<Editor: All orange text in this section is from TP4034a Dispersed Namespaces which is to only be included in the next numbered release>

…

If the namespace is not a dispersed namespace, then ~~T~~the size of the Reservation Status data structure is a function of the number of ~~controllers in the NVM subsystem that are associated with hosts that are~~ registrants of the namespace (i.e., there is a ~~Registered Controller~~ Registrant data structure and/or ~~Registered Controller~~ Registrant ~~e~~Extended data structure for each such ~~controller~~ registrant). If the namespace is a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems, then the size of the Reservation Status data structure is a function of the number of registrants of the namespace ~~controllers~~ in the NVM subsystem containing the controller processing the command ~~that are associated with hosts that are registrants of the namespace~~ and the number of ~~hosts that are~~ registrants of the namespace in each separate participating NVM subsystem.

For controllers compliant with NVM Express Base Specification, Revision 2.0 and earlier, registrants of the namespace that are not associated with any controller in the NVM subsystem may or may not be reported by this command.

…

Figure 404: Reservation Report – Command Dword 11

Bits	Description
31:~~01~~02	Reserved
01	**Dispersed Namespace Reservation Support (DISNSRS):** This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature). If this bit is set to ‘1’, then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a ~~Registered Controller~~Registrant data structure or ~~Registered Controller~~Registrant Extended data structure (refer to section 8.NEW.5)). If this bit is cleared to ‘0’, then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a ~~Registered Controller~~Registrant data structure or ~~Registered Controller~~Registrant Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to ‘0’, then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.
00	**Extended Data Structure (EDS):** If set to ‘1’, then the controller returns the extended data structure defined in Figure 406. If cleared to ‘0’, then the controller returns the data structure defined in Figure 405.

Figure 405: Reservation Status Data Structure

Bytes	Description
…	
06:05	**Number of ~~Registered Controllers (REGCTL)~~ Registrants (REGSTRNT):** This field indicates the number of ~~controllers that are associated with hosts that are~~ registrants of the namespace. This indicates the number of ~~Registered Controller~~ Registrant data structures ~~and/~~or ~~Registered Controller~~Registrant ~~e~~Extended data structures contained in this data structure. Note: This field was formerly named Number of Registered Controllers (REGCTL).
…	
47:24	**~~Registered Controller~~ Registrant Data Structure 0** Note: This field was formerly named Registered Controller Data Structure 0.
	...
24*n+47: 24*(n+1)	**~~Registered Controller~~ Registrant Data Structure n**

Figure 406: Reservation Status Extended Data Structure

Bytes	Description
…	
127:64	**~~Registered Controller~~ Registrant Extended Data Structure 0** Note: This field was formerly named Registered Controller Extended Data Structure 0.
	...
64*(n+1)+63: 64*(n+1)	**~~Registered Controller~~ Registrant Extended Data Structure n**

Figure 407: ~~Registered Controller~~Registrant Data Structure

Bytes	Description
01:00	**Controller ID (CNTLID):** If a registrant of the namespace is associated with a controller in the NVM subsystem, then t~~T~~h~~is~~ field contains the controller ID (i.e., the value of the CNTLID field in the Identify Controller data structure) of the controller ~~whose status is reported in this data structure~~ that is associated with the registrant whose host identifier is indicated in the Host Identifier field of this Registrant data structure. ~~If the controller is a dynamic controller (refer to section 3.1.1) that is not associated with a host, then the Controller ID field shall be set to FFFFh.~~ If a registrant of the namespace is not associated with any controller in the NVM subsystem, then the controller processsing the command shall set this field to FFFFh. If the namespace is a dispersed namespace and the controller is not contained in the same participating NVM subsystem as the controller processing the command, then the Controller ID field is set to FFFDh, as described in section 8.1.8.6.
02	**Reservation Status (RCSTS):** This field indicates the reservation status of the ~~controller described by this data structure~~ registrant associated with this data structure (i.e., the registrant whose host identifier is indicated in the Host Identifier field of this Registrant data structure). Bits 7:1 are reserved. Bit 0 is set to '1' if the ~~controller is associated with a host that~~ registrant associated with this data structure holds a reservation on the namespace.
07:03	Reserved
15:08	**Host Identifier (HOSTID):** This field contains the 64-bit Host Identifier of the ~~controller~~ registrant of the namespace described by this data structure.
23:16	**Reservation Key (RKEY):** This field contains the reservation key of the registrant~~ host associated with the controller~~ described by this data structure.

Figure 408: Registered Controller Extended Data Structure

Bytes	Description
…	
31:16	**Host Identifier (HOSTID):** This field contains the 128-bit Host Identifier of the ~~controller~~ registrant of the namespace described by this data structure.
63:32	Reserved

…

Modify section 8 as shown below:

8 Extended Capabilities

…

8.2 Boot Partitions

…

8.2.3. Boot Partition Protection

…

After activating Boot Partition Protection:

a) The default state for all Boot Partitions is the “Locked” state. In this state, host software may read a Boot Partition. In this state, the controller rejects attempts to write to a Boot Partition using the Firmware Commit command;

b) Each Boot Partition may be locked or unlocked independently using the corresponding bit in the Device Configuration Block data structure. A Boot Partition may be unlocked in the same command that enables Boot Partition Protection ~~Proteciton~~; and

c) …

…

8.19 Reservations

…

8.19.7 Preempting a Reservation or Registration

A host that is a registrant may preempt a reservation and/or registration by executing a Reservation Acquire command (refer to section 7.2 ~~7.1~~)~~, setting~~ specifying:

- the Reservation Acquire Action (RACQA) field set to 001b (i.e., Preempt) or set to 010b (i.e., Preempt and Abort);~~,~~ and ~~supplying~~
- the current reservation key associated with the host in the Current Reservation Key (CRKEY) field.

…

A host may abort commands as a side effect of preempting a reservation by executing a Reservation Acquire command (refer to section 7.1) and setting the RACQA field to 010b (Preempt and Abort). The behavior of such a command is exactly the same as that described above with the RACQA field set to 001b (Preempt), with two exceptions:

…

Description of NVM Express NVM Command Set Specification 1.0d changes

Modify section 2 as shown below:

2 NVM Command Set Model

…

2.1 Theory of operation

…

2.1.1 Namespaces

…

A logical block may be marked as allocated as the result of:

- a User Data Out Command not addressing the logical block (e.g., NPWG field may indicate sequential logical blocks placed and tracked together on the media (refer to section 5.8.2.1);
- a Write Zeroes command (refer to section 3.2.8) not addressing the logical block; ~~or~~
- a sanitize operation (refer to section 5.6); or
- a Format NVM command (refer to section 4.1.2).

~~A logical block may be marked deallocated as the result of~~Commands and operations that may result in a logical block being deallocated include:

- a Dataset Management command (refer to section 3.2.3);
- a Write Zeroes command (refer to section 3.2.8) addressing the logical block; ~~or~~
- a sanitize operation (refer to section 5.6); or
- a Format NVM command (refer to section 4.1.2).

Vendor specific means are able to allocate or deallocate logical blocks.

…

Modify section 4 as shown below:

4 Admin Commands for the NVM Command Set

4.1 Admin Command behavior for the NVM Command Set

…

4.1.5 Identify Command

…

4.1.5.1 NVM Command Set Identify Namespace Data Structure (CNS 00h)<Document change>

…

Figure 97: Identify – Identify Namespace Data Structure, NVM Command Set

Bytes	O/M [1]	Description
…		

Figure 97: Identify – Identify Namespace Data Structure, NVM Command Set

Bytes	O/M[1]	Description
…		
65:64	O	**Namespace Preferred Write Granularity (NPWG):** This field indicates the smallest recommended write granularity in logical blocks for this namespace. This is a 0's based value. If the OPTPERF bit is cleared to '0', then this field is reserved. The size indicated should be less than or equal to the maximum number of logical blocks that are able to be transferred based on the value of the Maximum Data Transfer Size (MDTS) field defined in the Identify Controller data structure (refer to the NVM Express Base Specification). The MDTS field ~~that~~ is specified in units of minimum memory page size. The value of this field may change if the namespace is reformatted. The size should be a multiple of Namespace Preferred Write Alignment (NPWA). Refer to section 5.8.2 for how this field is utilized to improve performance and endurance.
…		
73:72	O	**Namespace Optimal Write Size (NOWS):** This field indicates the size in logical blocks for optimal write performance for this namespace. This is a 0's based value. If the OPTPERF bit is cleared to '0', then this field is reserved. The size indicated should be less than or equal to the maximum number of logical blocks that are able to be transferred based on the value of the Maximum Data Transfer Size (MDTS) field defined in the Identify Controller data structure (refer to the NVM Express Base Specification). The MDTS field ~~that~~ is specified in units of minimum memory page size. The value of this field may change if the namespace is reformatted. The value of this field should be a multiple of Namespace Preferred Write Granularity (NPWG). If ~~the~~ this namespace is associated with an NVM set, ~~NOWS defined for this namespace~~ this field shall be set to the maximum number of logical blocks that are able to be transferred based on the value of the Optimal Write Size field setting defined in the NVM Set Attributes Entry (refer to the Namespace Identification Descriptor in the NVM Express Base Specification) for the NVM Set with which this namespace is associated. If NOWS is not supported, the Optimal Write Size field in NVM Sets Attributes Entry (refer to the Namespace Identification Descriptor in the NVM Express Base Specification) for the NVM Set with which this namespace is associated should be used by the host for I/O optimization. Refer to section 5.8.2 for how this field is utilized to improve performance and endurance.

…

4.2 I/O Command Set Specific Admin commands

…

4.2.1 Get LBA Status command

…

b) **Untracked LBAs:** a list of Potentially Unrecoverable LBAs generated by a scan originated by a Get LBA Status command with the ATYPE field set to 10h. The controller scans internal data structures related to the specified range of LBAs to determine which LBAs are Potentially Unrecoverable LBAs. The controller may use this scan to determine which LBAs in which namespaces are affected by a component (e.g., die or channel) failure. Significant delays may be incurred during the processing of a Get LBA Status command with the ATYPE field set to 10h. After the controller discovers~~y of~~ Untracked LBAs, ~~they~~ those LBAs may or may not be added to the list of Tracked LBAs.

…

Description of NVM Express Boot Specification 1.0 changes

Modify section 3 as shown below:

3 Boot Mechanisms

…

3.1 ACPI NVMe Boot Firmware Table (NBFT)

…

3.1.2 NBFT Structure

…

3.1.2.5 Subsystem Namespace (SSNS) Descriptor

…

Figure 15: Subsystem Namespace (SSNS) Descriptor

<table>
<tr><th>Bytes</th><th>O/M[1]</th><th>Description</th></tr>
<tr><td colspan="3">…</td></tr>
<tr><td>46</td><td>M</td><td><b>Primary HFI Descriptor Index:</b> This field indicates the value of the HFI Descriptor Index field of the HFI Descriptor (refer to Figure 11) for the interface associated with this namespace. If multiple HFIs are associated with this record, subsequent interfaces should be populated in the Secondary HFI Associations field (refer to section A.4.1.x).</td></tr>
<tr><td>47</td><td></td><td>Reserved</td></tr>
<tr><td>53:48</td><td>O</td><td><b>Secondary HFI Associations Heap Object Reference:</b> If this field is set to a non-zero value, then:
a) this field indicates an array of bytes, in which each byte contains the value of the HFI Descriptor Index field of an HFI Descriptor in the HFI Descriptor List (refer to Figure 10 and section A.4.1.x); and
b) the Offset field and the Length field shall be set to non-zero values.
If this field is cleared to 0h, then no secondary HFI associations are specified.
<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>03:00</td><td><b>Offset:</b> Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header.</td></tr>
<tr><td>05:04</td><td><b>Length:</b> Length in bytes of the heap object, if any.</td></tr>
</table>
</td></tr>
<tr><td colspan="3"><b>…</b></td></tr>
<tr><td colspan="3">Notes:
1. O/M definition: O = Optional, M = Mandatory.</td></tr>
</table>

…

3.1.2.5.5 Subsystem and Namespace Extended Information Descriptor

…

Figure 19: Subsystem and Namespace Extended Information Descriptor

Bytes	O/M[1]	Description
…		
17:12	O	**DHCP Root Path String Heap Object Reference:** If the SSNS DHCP Root-Path Override F~~f~~lag bit is set to '1', then:

Figure 19: Subsystem and Namespace Extended Information Descriptor

<table>
<tr><th>Bytes</th><th>O/M[1]</th><th>Description</th></tr>
<tr><td></td><td></td><td>a) this field indicates the offset in bytes of a heap object containing an DHCP Root Path String as defined in section 3.1.2.5.3 used by the driver.
b) the Offset field and the Length field shall be set to non-zero values.

If the SNSS DHCP Root-Path Override F~~f~~lag bit is cleared to ’0’, then this field is reserved.
<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>03:00</td><td><b>Offset:</b> Offset in bytes of the heap object, if any, from byte offset 0h of the NBFT Table Header.</td></tr>
<tr><td>05:04</td><td><b>Length:</b> Length in bytes of the heap object, if any.</td></tr>
</table>
</td></tr>
<tr><td colspan="3">Notes:
1. O/M definition: O = Optional, M = Mandatory.</td></tr>
</table>

…

Modify section Appendix A as shown below:

Annex A. Booting (Informative)

…

A.1 Domain Specific Terms

…

A.1.4 UEFI Device Handle

…

A.1.4.x Usage of Primary and Secondary HFI fields in the SNSS

The Secondary HFI Associations Heap Object Reference field in the SNSS provides a mechanism to associate more than one HFI for each SNSS record in an NBFT. If a pre-OS driver needs to describe two or more HFIs that represent different physical or logical interfaces with otherwise common connectivity to a common SNSS parameters, then the driver may choose to represent that connectivity by either:

- creating additional SNSS Descriptor records that only utilize a Primary HFI Descriptor Index; or
- utilizing the SNSS Secondary HFI Associations Heap Object Reference.

A driver may choose to create new SNSS Descriptor records per HFI if:

- there are any path, channel, or transport differences; or
- a pre-OS driver does not differentiate between path types.

Figure X.1 shows an example of a subset of fields in an NBFT, where a driver creates additional SNSS records associated with the HFI. In this example, there are two HFI Descriptors (HFI Descriptor Index 03h and HFI Descriptor Index 04h) that represent two interfaces on a common Layer 3 TCP/IP subnet. These two interfaces, described by HFI Descriptor Index 03h and 04h can both communicate, with:

- common reachability to different SNSS Descriptors (Index 5 and Index 6); and
- an empty Secondary HFI Associations Heap Object Reference field.

Figure X.1: Example NBFT with two SNSS records with two HFIs

Descriptor	Bytes	Field	Value
HFI Descriptor Index 03h			
HFI Descriptor	00	**Structure ID**	03h
	01	**HFI Descriptor Index**	03h

Figure X.1: Example NBFT with two SNSS records with two HFIs

Descriptor	Bytes	Field	Value
	…		
HFI Transport Info Descriptor – NVMe/TCP	00	**Structure ID**	07h
	01	**Version**	01h
	02	**HFI Transport Type**	05h
	03	**Transport Info Version**	02h
	05:04	**HFI Descriptor Index**	00h 03h
	…		
	35:20	**IP Address**	c0h a8h 01h 03h
	36	**Subnet Mask Prefix**	18h
	52:37	**IP Gateway**	c0h a8h 01h feh
	…		
HFI Descriptor Index 04h			
Host Fabric Interface Descriptor	00	**Structure ID**	03h
	01	**HFI Descriptor Index**	04h
	…		
HFI Transport Info Descriptor – NVMe/TCP	00	**Structure ID**	07h
	01	**Version**	01h
	02	**HFI Transport Type**	05h
	03	**Transport Info Version**	02h
	05:04	**HFI Descriptor Index**	00h 04h
	…		
	35:20	**IP Address**	c0h a8h 01h 04h
	36	**Subnet Mask Prefix**	18h
	52:37	**IP Gateway**	c0h a8h 01h feh
	…		
SNSS Record (Index 5)			
Subsystem Namespace (SNSS) Descriptor	00	**Structure ID**	04h
	02:01	**SNSS Descriptor Index**	00h 05h
	…		
	46	**Primary HFI Descriptor Index**	03h
	53:48	**Secondary HFI Associations Heap Object Reference**	00h
	…		
SNSS Record (Index 6)			
Subsystem Namespace (SNSS) Descriptor	00	**Structure ID**	04h
	02:01	**SNSS Descriptor Index**	00h 06h
	…		
	46	**Primary HFI Descriptor Index**	04h
	53:48	**Secondary HFI Associations Heap Object Reference**	00h
	…		

By utilizing the SNSS Secondary HFI Associations Heap Object Reference, the driver or administrator may optimize space or indicate multipath connectivity. Figure X.2 shows this example using a subset of NBFT fields with two HFI Descriptors (Structure ID 1 and Structure ID 2) that represent common Layer 3 TCP/IP subnet interfaces, both with common reachability to the same SNSS Descriptor (Index 1).

Figure X.2: Example SNSS with two HFIs

Descriptor	Bytes	Field	Value
HFI Descriptor Index 01h			
Host Fabric Interface Descriptor	00	**Structure ID**	03h
	01	**HFI Descriptor Index**	01h
	…		
HFI Transport Info Descriptor – NVMe/TCP	00	**Structure ID**	07h
	01	**Version**	01h
	02	**HFI Transport Type**	05h
	03	**Transport Info Version**	02h
	05:04	**HFI Descriptor Index**	00h 01h
	…		
	35:20	**IP Address**	c0h a8h 01h 01h
	36	**Subnet Mask Prefix**	18h
	52:37	**IP Gateway**	c0h a8h 01h feh
	…		
HFI Descriptor Index 02h			
Host Fabric Interface Descriptor	00	**Structure ID**	03h
	01	**HFI Descriptor Index**	02h
	…		
HFI Transport Info Descriptor – NVMe/TCP	00	**Structure ID**	07h
	01	**Version**	01h
	02	**HFI Transport Type**	05h
	03	**Transport Info Version**	02h
	05:04	**HFI Descriptor Index**	00h 02h
	…		
	35:20	**IP Address**	c0h a8h 01h 02h
	36	**Subnet Mask Prefix**	18h
	52:37	**IP Gateway**	c0h a8h 01h feh
	…		
SNSS Record (Index 1)			
Subsystem Namespace (SNSS) Descriptor	00	**Structure ID**	04h
	02:01	**SNSS Descriptor Index**	00h 01h
	…		
	46	**Primary HFI Descriptor Index**	01h
	48:53	**Secondary HFI Associations Heap Object Reference**	**Bytes** **Field** **Value** 03:00 **Offset** 400h 05:04 **Length** 01h
	…		
Secondary HFI Associations Heap Object [Offset 400h]	03:00	n/a	02h

…

Description of NVM Express Management Interface Specification 1.2d changes

Modify section 4 as shown below:

4 Message Servicing Model

…

4.1 NVMe-MI Messages

…

4.1.2 Response Messages

…

The format of the Response Body is dependent on the Response Message Status. Figure 28 references the section that defines the format of the Response Message for each Response Message Status value. If multiple error Response Message Status values apply ~~error conditions are present~~, then the ~~a~~ Responder ~~may~~ selects one of those applicable Response Message Status values ~~choose which error status~~ to report.

Figure 28: Response Message Status Values

Value	Description	Response Message Format Section
…		
05h	**Invalid Command Size:** The size of the Message Body of the Request Message was different than expected due to a reason other than the Command Message requiring Request Data and containing too much or too little Request Data (e.g., the Request Message did not contain all the required parameters ~~or Request Data was present when not expected)~~. The expected size of the Message Body is determined by the NVMe-MI Message Type and opcode assuming no other errors are detected (e.g., Invalid Command Opcode or Invalid Parameter).	4.1.2.1
06h	**Invalid Command Input Data Size:** The Command Message requires Request Data and contains too much or too little Request Data.	4.1.2.1
…		

…

Modify section 5 as shown below:

5. Management Interface Command Set

…

5.4 Management Endpoint Buffer Read

…

If the Data Offset (DOFST) field is greater than or equal to the size of the Management Endpoint Buffer, then the Management Endpoint shall respond~~s~~ with an Invalid Parameter Error Response with the PEL field indicating the DOFST field. If the DOFST field is less than the size of the Management Endpoint Buffer and the sum of the DOFST and DLEN fields is greater than ~~or equal to~~ the size of the Management Endpoint

Buffer, then the Management Endpoint shall respond~~s~~ with an Invalid Parameter Error Response with the PEL field indicating the DLEN field.

When an attempt is made to read Management Endpoint Buffer contents that were zeroed due to a sanitize operation, then the Management Endpoint shall respond~~s~~ with a Response Message Status of Management Endpoint Buffer Cleared Due to Sanitize.

…

5.5 Management Endpoint Buffer Write

…

If the Data Offset (DOFST) field is greater than or equal to the size of the Management Endpoint Buffer, then the Management Endpoint shall respond~~s~~ with an Invalid Parameter Error Response with the PEL field indicating the DOFST field. If the DOFST field is less than the size of the Management Endpoint Buffer and the sum of the DOFST and DLEN fields is greater than ~~or equal to~~ the size of the Management Endpoint Buffer, then the Management Endpoint shall responds with an Invalid Parameter Error Response with the PEL field indicating the DLEN field.

…

5.12 VPD Read

The VPD Read command is used to read the Vital Product Data described in section 8.2. Upon successful completion of the VPD Read command, the specified portion of the VPD contents is returned in the Response Data.

The VPD Read command uses NVMe Management Dword 0 and 1. The format of NVMe Management Dwords 0 and 1 are shown in Figure 110 and Figure 111 respectively. There is no Request Data sent in the Request Message.

If a ~~A~~ VPD Read command with the DLEN field cleared to 0h is processed, ~~length 0 and no data is valid. The~~ then the Responder shall respond~~s~~ with a Success Response and no Response Data. ~~If the Data Length plus Data Offset fields are greater than the size of the VPD, then the Responder does not return the VPD contents and responds with an Invalid Parameter Error Response with the PEL field indicating the Data Length field.~~ If the Data Offset (DOFST) field is greater than or equal to the maximum size of the FRU Information Device, then the Management Endpoint should respond with an Invalid Parameter Error Response with the PEL field indicating the DOFST field. If the DOFST field is less than the maximum size of the FRU Information Device and the sum of the DOFST and DLEN fields is greater than the maximum size of the FRU Information Device, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating the DLEN field.

<Note to editor: Change the “should” in the first sentence of the preceding paragraph to a “shall” in the non-errata version of the integration spec.>

…

5.13 VPD Write

…

The Requester should not read the contents of the VPD while this command is servicing. Reading the contents of the VPD or the processing of a VPD Read command while a VPD Write command is being processed may return incorrect data as a result of the read.

~~If the Data Length plus Data Offset fields are greater than the size of the VPD, then the Responder does not write to the VPD and responds with an Invalid Parameter Error Response with the PEL field indicating the Data Length field.~~ If the Data Offset (DOFST) field is greater than or equal to the maximum size of the

FRU Information Device, then the Management Endpoint should not write the contents of the VPD and should respond with an Invalid Parameter Error Response with the PEL field indicating the DOFST field. If the DOFST field is less than the maximum size of the FRU Information Device and the sum of the DOFST and DLEN fields is greater than the maximum size of the FRU Information Device, then the Management Endpoint shall not write the contents of the VPD and shall respond with an Invalid Parameter Error Response with the PEL field indicating the DLEN field.

<Note to editor: Change the two instances of “should” in the first sentence of the preceding paragraph to a “shall” in the non-errata version of the integration spec.>

…

Modify section 6 as shown below:

6 NVM Express Admin Command Set

…

Figure 118: NVMe Admin Command Request Description

Bytes	Description
…	…

Figure 118: NVMe Admin Command Request Description

Bytes	Description
31:28	**Data Offset (DOFST):** ~~For commands that transmit data from the Management Controller to the Management Endpoint (i.e., the Data Transfer subfield for the opcode field of the NVMe Admin Command as defined by the NVM Express Base Specification is 01b) or do not transmit data (i.e., the Data Transfer subfield in the opcode field of the NVMe Admin Command as defined by the NVM Express Base Specification is 00b), this field should be cleared to 0h. If this field is not 0h for commands that transmit data from the Management Controller to the Management Endpoint or that do not transfer data, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ T~~t~~his field specifies the starting offset, in bytes, of the portion of data contained in the NVMe Admin Command completion data that ~~is~~ shall be returned starting at byte offset 0h of the NVMe Response Data field in the Response Message f~~F~~or commands that ~~transmit~~ are defined to transfer NVMe Response ~~d~~Data from ~~the~~ a Management Endpoint to ~~the~~ a Management Controller ~~(i.e., the Response Data field in the Response Message has non-zero length),~~ ~~this field specifies the starting offset, in bytes, of the portion of data contained in the NVMe Admin Command completion data that is returned starting at byte offset 0h of the Response Data field in the Response Message~~. This field should be cleared to 0h for all other commands (i.e., commands that are not defined to transmit data and commands that are defined to transfer NVMe Request Data from a Management Controller to a Management Endpoint). ~~This field should be less than the size of the NVMe Admin Command completion data. If this field is greater than or equal to the size of the NVMe Admin Command completion data, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ ~~Bits 1:0 of this field should be cleared to 00b. If bits 1:0 are not cleared to 00b, then the Management Endpoint should respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ The Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field if any of the following conditions are true: • the value of the DOFST field is greater than 0h for commands that are not defined to transfer NVMe Response Data; or • the value of the DOFST field is greater than or equal to the size of the NVMe Admin Command completion data. The Management Endpoint should respond with an Invalid Parameter Error Response with the PEL field indicating this field if any of the following conditions are true: • bits 1:0 of the DOFST field are not cleared to 00b. <Note to Editor: In the non-errata integration copy of the specification, delete the prior sentence and move "bits 1:0 of the DOFST field are not cleared to 00b" to the prior list starting with "The Management Endpoint shall respond".>

Figure 118: NVMe Admin Command Request Description

Bytes	Description
35:32	**Data Length (DLEN):** ~~For commands that do not transfer data (i.e., the Data Transfer subfield in the opcode field of the NVMe Admin Command as defined by the NVM Express Base Specification is 00b), this field should be cleared to 0h. If this field is not 0h for commands that do not transfer data, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ For commands that~~ transmit~~ are defined to transfer Request ~~d~~Data from ~~the ~~a Management Controller to ~~the ~~a Management Endpoint~~ (i.e., the Data Transfer subfield in the opcode field of the NVMe Admin Command as defined by the NVM Express Base Specification is 01b)~~, this field specifies the length, in bytes, of the data contained in the NVMe Request Data field in the Request Message. For commands that ~~transmit~~ are defined to transfer Response ~~d~~Data from ~~the ~~a Management Endpoint to ~~the ~~a Management Controller~~ (i.e., the Data Transfer subfield in the opcode field of the NVMe Admin Command as defined by the NVM Express Base Specification is 10b)~~, this field indicates ~~specifies ~~the length, in bytes, of the portion of data contained in the NVMe Admin Command completion data that ~~is ~~shall be returned in the NVMe Response Data field in the Response Message. ~~The sum of DLEN plus DOFST should be less than or equal to the size of the NVMe Admin Command completion data~~. ~~If the sum is greater, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ ~~For commands that transmit data (i.e., the Data Transfer subfield in the opcode of the NVMe Admin Command as defined by the NVM Express Base Specification is 01b or 10b), the Management Controller should specify a non-zero length in this field. If this field is cleared to 0h for commands that transmit data, then the Management Endpoint should respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ ~~Bits 1:0 of this field should be cleared to 00b. If bits 1:0 are not cleared to 00b, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ ~~This field should be less than or equal to 4,096. If this field is greater than 4,096, then the Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field.~~ The Management Endpoint shall respond with an Invalid Parameter Error Response with the PEL field indicating this field if any of the following conditions are true: • bits 1:0 of the DLEN field are not cleared to 00b; • the value of the DLEN field is greater than 0h for commands that are not defined to transfer NVMe Request Data or NVMe Response Data; • the value of the DLEN field is greater than 4,096; or • the sum of the value of the DLEN field plus the value of the DOFST field is greater than the size of the NVMe Admin Command completion data for commands that are defined to transfer NVMe Response Data. The Management Endpoint should respond with an Invalid Parameter Error Response with the PEL field indicating this field if any of the following conditions are true: • the value of the DLEN field is not equal to the length of the NVMe Request Data field required by the command; or • the DLEN field is cleared to 0h for commands that are defined to transfer NVMe Request Data or NVMe Response Data. <Note to Editor: In the non-errata integration copy of the specification, delete the prior sentence and move the two bullet points in the prior sentence to the list starting with “The Management Endpoint shall respond”.>
…	**…**

…

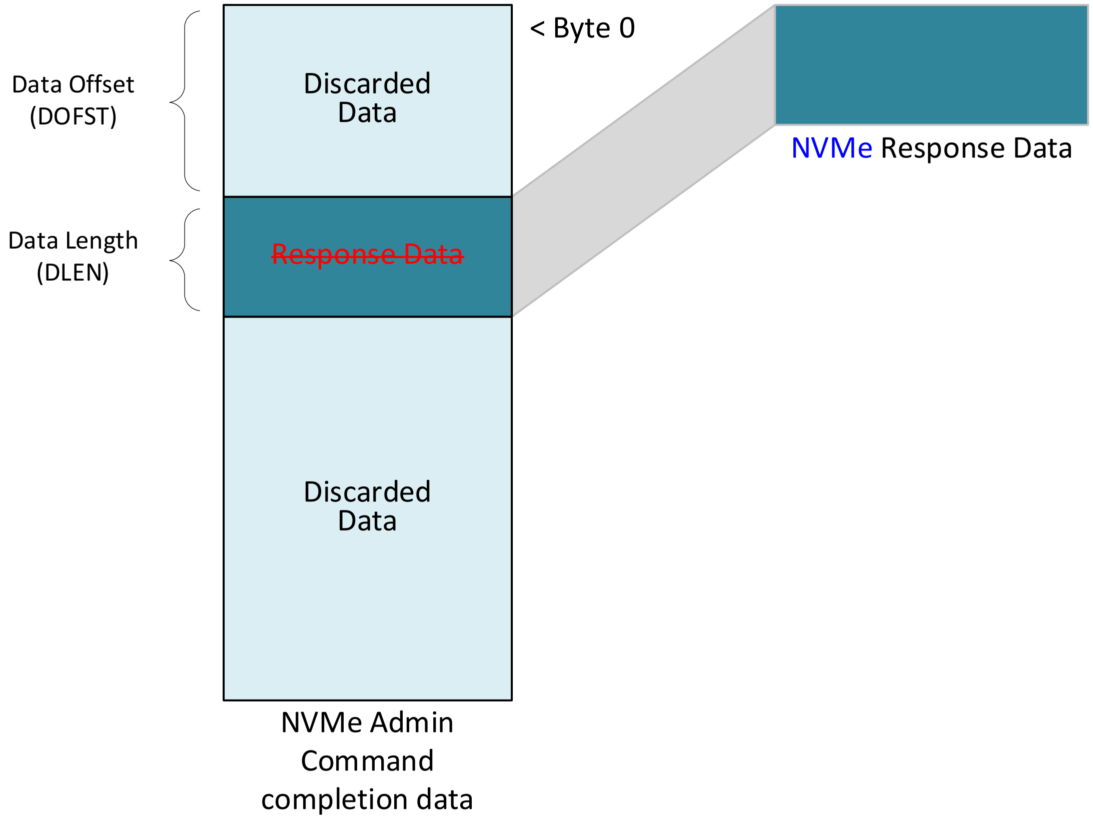
6.1 Request and Response Data

NVMe Admin Commands sent using the out-of-band mechanism may contain data as part of the Command Message. This data is passed in the NVMe Request Data field (refer to Figure 117) instead of using PRP

Lists or SGL segments. The PRP Entry 2 (PRP2) and Metadata Pointer (MPTR) fields within ~~the~~ NVMe Admin Commands sent using the out-of-band mechanism are reserved.

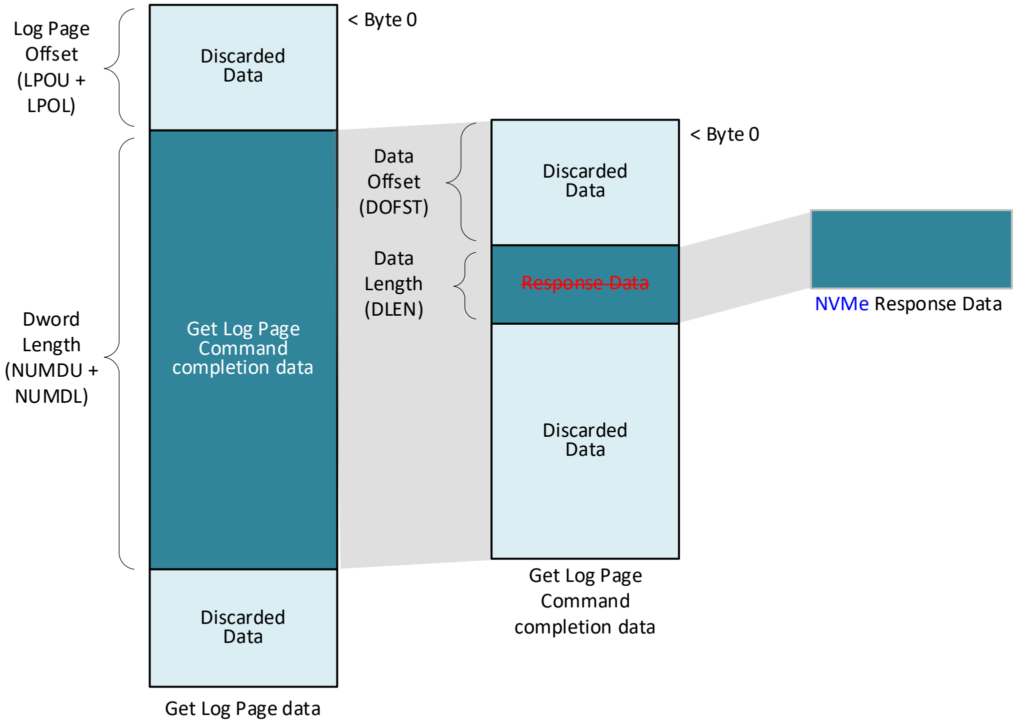
If ~~there is~~ the NVMe Admin Command is defined to transfer data in the NVMe Response Data field (refer to Figure 119) when sent using the out-of-band mechanism ~~expected in the Response Message in the completion of the NVMe Admin Command (i.e., the Data Transfer subfield in the corresponding NVMe Admin Command for the opcode is 10b)~~, then the Data Offset and Data Length fields describe the portion of the NVMe Admin Command completion data that is transferred in the NVMe Response Data field ~~Response Message~~. Any remaining data not transferred in the NVMe Response Data field ~~Response Message~~ is discarded by the Management Endpoint as shown in Figure 121.

Figure 121: NVMe Admin Command Response Data Example



Some NVMe Admin Commands specify an offset and length which shall be applied first to create the NVMe Admin Command completion data. Then DOFST and DLEN shall be applied to that Admin Command completion data which may further reduce the amount of NVMe Response Data. Figure 122 provides an example for the Get Log Page command.

Figure 122: NVMe Get Log Page Command Response Data Example



…

6.2 Status

A Response Message for an NVMe Admin Command may contain two status fields. The first status field, contained in Byte 4 of the Response Message, is defined by this specification, and the second Status field, if present, is contained in the Completion Queue Entry Dword 3 field and defined in the NVM Express Base Specification.

An NVMe Admin Command Request Message is well formed if ~~it~~ that Request Message does not result in ~~contain~~ any of the following reported errors:

- Invalid Command Opcode (e.g., the opcode is not listed in Figure 116);
- Invalid Parameter (e.g., the Controller ID field specifies a Controller ID not implemented in the NVM Subsystem);
- Invalid Command Size (e.g., the Request Message does not contain a complete command); or
- Invalid Command Input Data Size (e.g., the Request Data field is larger than the size specified in the Data Length field).

If the NVMe Admin Command Request Message is well formed, then a Success Response shall be ~~is~~ transmitted. The Success Response shall contain~~s~~ the status associated with NVMe Admin Command in the Status field of the Completion Queue Entry Dword 3 field. The Status field shall contain~~s~~ any NVM Express Base Specification and I/O Command Set specifications specific status codes (e.g., Success or Invalid Parameter).

…

Modify section 7 as shown below:

7 PCIe Command Set (Optional)

…

A PCIe Command that is not well-formed results in an Error Response. A PCIe Command is well formed if that PCIe command ~~it~~ does not ~~contain~~ result in any of the following reported errors:

- Invalid Command Opcode (e.g., the Opcode is not listed in Figure 130);
- Invalid Parameter (e.g., the Controller ID field specifies a Controller ID not implemented in the NVM Subsystem);
- Invalid Command Size (e.g., the Request Message does not contain a complete command); or
- Invalid Command Input Data Size (e.g., the NVMe Request Data field is larger than the size expected by the command).

…