LEGAL NOTICE:

LEGAL DISCLAIMER:

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

NVM Express™ Technical Errata

Errata ID	**001**
Revision Date	**3/9/2020**
Affected Spec Ver.	**NVM Express™ 1.4**
Corrected Spec Ver.	

Errata Author(s)

Name	**Company**
Mike Allison	Intel
Fred Knight	NetApp Inc.
Paul Suhler	Micron
David Black	DellEMC
Judy Brock	Samsung
Fred Au	Marvell
Meghvendra Rathod	Cadence
Prashanth Adurthi	NetApp
Brad Besmer	Broadcom
John Geldman	Toshiba Memory
Yoni Shternhell	Western Digital Technologies, Inc.
Curtis Ballard	HPE

Errata Overview

- Made the use of Admin Queue and I/O Queue consistent
- Clarified the reset term in the definition of terms and abbreviations (HwInit and Reset)
- Added the definition of host-accessible memory
- Clarified the definition of metadata
- Clarified the reserved keyword definition
- Aligned to PCI Express® Base Specification, Revision 4.0
- Corrected the offset in : BAR3 – Vendor Specific
- Corrected the Message Signaled Interrupt Pending Bits register access type.
- Corrected the PCI Express Device Control register reset values.
- Corrected the AER Correctable Error Mask register reset values.
- Clarified that CQDA field memory alignment and SQE and CQE granularity size restrictions
- Clarified restrictions in the command CID field use of the value FFFFh
- Clarified PRP Lists referenced memory does not have to be physically contiguous memory pages
- Removed conflicting dword alignment of Metadata Region
- Removed redundant SGL Segment descriptor in the Invalid SGL Segment Descriptor description
- Clarified when the Controller Memory Buffer is undefined to the host
- Clarified out of scope usage for the Persistent Memory Region
- Clarified a recommendation on the PC field in the Create I/O Completion Queue command
- Updated the completion sections of all commands that may issue Namespace Write Protected status code
- Removed the conflict of using Overlapping Range status code on failure to conform to FWUG field
- Clarified optional command dwords for Get Features and Set Features commands
- Clarified that a SMART / Health Information log page has either controller or namespace scope in Figure 191 to match the loge page definition
- Clarified the definition of the Phase Tag bit 0 in Error Log entry
- Clarified that Write Zeroes commands do not impact Data Units Written field
- Clarified byte offsets in Persistent Event Log Events
- Corrected the NCAP field size in the Change Namespace Event
- Clarified references to Format NVM and Set Features commands
- Clarified the definition of Vendor Specific Event Data field in the Vendor Specific Event Descriptor Format
- Moved thin provisioning and the definition of when a LBA is deallocated to a new section
- Clarified when fields in Identify Namespace data structure are reserved
- Removed conflicting status codes for Compare and Write fused operation
- Clarified Namespace Insufficient Capacity field definition
- Clarified that allocated namespaces formatted in Format NVM command
- Clarified Format NVM command aborting
- Clarified when fields are ignored
- Clarified the definition of values read from a deallocated LBA
- Clarified field names in Write Zeroes command
- Removed redundant commands listed for I/O controller support
- Updated the Controller Level Reset methods to align with PCIe specification
- Clarified restrictions of a reverse domain name in an NQN
- Clarified aborting a command targeting Type 3 protection namespace
- Clarified user of NVM storage of Boot Partitions

Revision History

Revision Date	Change Description
7/9/2019	Initial creation
7/10/2019	Fixed repeated commands in tables. Added entire list to Controller Level Reset. Made corrections from 7/11/2019 Technical WG meeting. Removed update to section 8.1 as a change to section 7.3.2 was required to move DL_Down to Conventional Reset to agree with PCIe 4.0.
7/26/2019	Corrections due to review in Technical WG meeting
8/1/2019	Corrections due to review in Technical WG meeting
8/2/2019	Correct the NCAP field size in the Changed Namespace Event. Added clarity yo variable byte offsets in Persistent Event Logs events.
8/14/2019	Corrected the scope of SMART / Health Log Page in Figure 191
9/12/2019	Moved the LBA allocation and deallocation definition to its own section from the ZNS group. Added names to the NSFEAT field bits.
9/20/2019	Added register space updates based on Controller Level Reset and Controller Reset that was accidently removed from NVMe 1.4 in section 7.3.2.
10/3/2019	Editorial changes
10/10/2019	Moved text to use if..then structure. Bounded the timing of CMB contents being undefined. Added changes to section 5.4. Aligned sections to be sequential. Editorial edits on the CWD usage for Get/Set Features commands. Editorial changes on the Format NVM command restrictions. Removed the TCG invalid security state from the Invalid Format status code in Figure 329. Updated section 6.1.TBD to use the list format instead of sentences.
10/15/2019	Updated per deallocation text provided by Christoph Wellwig.
10/17/2019	Fixed the TCG wording for Format NVM status codes.
10/21/2019	Aligned CQDA field to section 4.1.3 on sizing to slots. Updated the clarity of Format NVM command handling.
10/31/2019	Clarified the definition of reserved.
11/1/2019	Replaced the word “which” to the word “that”. Updated the errata overview to include the keyword reserved definition.
11/18/2019	Modified CQDA to only allow Dword alignment when CDW11.PC is set to ‘1’ for a Create I/O Submission Queue command and Create I/O Completion Queue command.
11/19/2019	In the ZNS meeting Jim Hatfield pointed out that “k” means Kilo so I updated a figure that used “k” that does not mean Kilo.
11/21/2019	Technical WG agreement to go to 30-day member review. Removed comments and formatting changes.
1/13/2020	Editorial changes on THINP and Namespace Insufficient Capacity definitions.
3/9/2020	Ratified

Incompatible Changes

- The CMB Queue Dword Alignment (CQDA) filed in the Controller Memory Buffer Location register added the requirement that size granularity of the physically contiguous memory shall be the size of SQE or CQE.
- The Namespace Capacity field in the Change Namespace Event Data Format (refer to Figure 222) was corrected to be 8B.

- The reset value of fields in the AER Correctable Error Mask register changes to reflect the PCIe Base Specification.
- A recommendation is added that the value of FFFFh for a SQE CID field should not be used as this value is reserved by the Error Information log page to indicate the error is not associated with a command.

Description of Specification Changes

Modify a portion of section 1.5 as shown below:

1.5 Conventions

…

Inside the register sections (i.e., section **Error! Reference source not found.** and section **Error! Reference source not found.**), the following terms and abbreviations are used:

RO	Read Only
RW	Read Write
R/W	Read Write. The value read may not be the last value written.
RWC	Read/Write '1' to clear
RWS	Read/Write '1' to set
Impl Spec	Implementation Specific – the controller has the freedom to choose its implementation.
HwInit	The default state is dependent on NVM Express controller and system configuration. ~~The value is initialized at reset, for example by an expansion ROM, or in the case of integrated devices, by a platform BIOS.~~
Reset	For section 2, this ~~This~~ column indicates the value of the field after a reset as defined by the appropriate PCI or PCI Express specifications. For section 3, this column indicates the value of the field after a Controller Level Reset as defined in section 7.3.2.

Add a new section 1.6.TBD as shown below:

1.6.TBD host-accessible memory

Memory that the host is able to access (e.g., host memory, Controller Memory Buffer (CMB), Persistent Memory Region (PMR)).

Modify a portion of section 1.6.25 as shown below:

1.6.25 metadata

Metadata is contextual information ~~about~~ related to a particular LBA of data. The host may include metadata to be stored by the NVM subsystem if storage space is provided by the controller. Metadata may include Protection Information (refer to section **Error! Reference source not found.**).

Modify a portion of section 1.7.5 as shown below:

1.7.5 reserved

A keyword referring to bits, bytes, words, fields, and opcode values that are set-aside for future standardization. Their use and interpretation may be specified by future extensions to this or other specifications. A reserved bit, byte, word, field, or register shall be cleared to 0h, or in accordance with a future extension to this specification. The recipient of a command

or a register write is not required to check reserved bits, bytes, words, or fields. Receipt of reserved coded values in defined fields in commands shall be reported as an error. Writing a reserved coded value into a controller register field produces undefined results.

Modify a portion of section 1.9 as shown below:

1.9 References

…
PCI Express® Base Specification, Revision 4.0~~3.1a~~. Available from http://www.pcisig.com.

Modify a portion of section 2.1.13 as shown below:

2.1.13 Offset 1Ch to ~~20~~1Fh: BAR3 – Vendor Specific

The BAR3 register is vendor specific. Vendor specific space may also be allocated at the end of the memory registers defined in section 3.

Modify a portion of Figure 41 in section 2.3.7 as shown below:

2.3.7 Offset MSICAP + 14h: MPEND – Message Signaled Interrupt Pending Bits (Optional)

…

Figure 1: Offset MSICAP + 14h: MPEND – Message Signaled Interrupt Pending Bits (Optional)

Bits	Type	Reset	Description
31:00	R~~W~~O	0h	**Pending Bits (PEND):** For each Pending bit that is set to '1', the function has a pending associated message.

Modify a portion of Figure 51 in section 2.5.4 as shown below:

2.5.4 Offset PXCAP + 8h: PXDC – PCI Express Device Control

…

Figure 2: Offset PXCAP + 8h: PXDC – PCI Express Device Control

Bits	Type	Reset	Description
15	RW	0b	**Initiate Function Level Reset:** A write of '1' initiates Function Level Reset to the Function. The value read by software from this bit shall always '0'.
14:12	RW/ RO	Impl Spec	**Max_Read_Request_Size (MRRS):** This field sets the maximum Read Request size for the Function as a Requester. The Function shall not generate Read Requests with size exceeding the set value.
11	RW/ RO	~~0b~~ Impl Spec	**Enable No Snoop (ENS):** If this bit is set to '1', the Function is permitted to set the No Snoop bit in the Requestor Attributes of transactions it initiates that do not require hardware enforced cache coherency. This bit may be hardwired to '0' if a Function would never set the No Snoop attribute in transactions it initiates.

Figure 2: Offset PXCAP + 8h: PXDC – PCI Express Device Control

Bits	Type	Reset	Description
10	RW/ RO	0b	**AUX Power PM Enable (APPME):** If this bit is set to '1', enables a Function to draw AUX power independent of PME AUX power. Functions that do not implement this capability hardware this bit to '0'.
09	RW/ RO	0b	**Phantom Functions Enable (PFE):** If this bit is set to '1', enables a Function to use unclaimed Functions as Phantom Functions to extend the number of outstanding transaction identifiers. If this bit is cleared to '0', the Function is not allowed to use Phantom Functions.
08	RW/ RO	0b	**Extended Tag Enable (ETE):** If this bit is set to '1', enables a Function to use an 8-bit Tag field as a Requester. If this bit is cleared to '0', the Function is restricted to a 5-bit Tag field.
07:05	RW/ RO	000b	**Max_Payload_Size (MPS):** This field sets the maximum TLP payload size for the Function. As a receiver, the Function shall handle TLPs as large as the set value. As a transmitter, the Function shall not generate TLPs exceeding the set value. Functions that support only the 128 byte max payload size are permitted to hardwire this field to 0h.
04	RW/ RO	Impl Spec	**Enable Relaxed Ordering (ERO):** If this bit is set to '1', the Function is permitted to set the Relaxed Ordering bit in the Attributes field of transactions it initiates that do not require strong write ordering.
03	RW	0b	**Unsupported Request Reporting Enable (URRE):** This bit, in conjunction with other bits, controls the signaling of Unsupported Requests by sending error messages.
02	RW	0b	**Fatal Error Reporting Enable (FERE):** This bit, in conjunction with other bits, controls the signaling of Unsupported Requests by sending ERR_FATAL messages.
01	RW	0b	**Non-Fatal Error Reporting Enable (NFERE):** This bit, in conjunction with other bits, controls the signaling of Unsupported Requests by sending ERR_NONFATAL messages.
00	RW	0b	**Correctable Error Reporting Enable (CERE):** This bit, in conjunction with other bits, controls the signaling of Unsupported Requests by sending ERR_COR messages.

Modify a portion of Figure 64 in section 2.6.6 as shown below:

2.6.6 Offset AERCAP + 14h: AERCEM – AER Correctable Error Mask Register

…

Figure 3: Offset AERCAP + 14h: AERCEM – AER Correctable Error Mask Register

Bits	Type	Reset	Description
31:16	RO	0h	Reserved
15	RW/RO	1~~0~~b	**Header Log Overflow Mask (HLOM)** (Optional)
14	RW/RO	1~~0~~b	**Corrected Internal Error Mask (CIEM)** (Optional)
13	RW	1~~0~~b	**Advisory Non-Fatal Error Mask (ANFEM)**
12	RW	0b	**Replay Timer Timeout Mask (RTM)**
11:09	RO	000b	Reserved
08	RW	0b	**REPLAY_NUM Rollover Mask (RRM)**
07	RW	0b	**Bad DLLP Mask (BDM)**
06	RW	0b	**Bad TLP Mask (BTM)**
05:01	RO	0h	Reserved
00	RW	0b	**Receiver Error Mask (REM)**

Modify a portion of Figure 84 in section 3.1.11 as shown below:

3.1.11 Offset 38h: CMBLOC – Controller Memory Buffer Location

…

Figure 4: Offset 38h: CMBLOC – Controller Memory Buffer Location

Bits	Type	Reset	Description
31:12	RO	Impl Spec	**Offset (OFST):** Indicates the offset of the Controller Memory Buffer in multiples of the Size Unit specified in CMBSZ.
11:09	RO	000b	Reserved
08	RO	Impl Spec	**CMB Queue Dword Alignment (CQDA):** ~~If this bit is set to '1', then I/O Submission Queues and I/O Completion Queues contained in the Controller Memory Buffer are dword aligned. If this bit is cleared to '0', then the I/O Submission Queues and I/O Completion Queues contained in the Controller Memory Buffer are aligned as defined by the PRP1 field of a Create I/O Submission Queue command (refer to **Error! Reference source not found.**) or a Create I/O Completion Queue command (refer to **Error! Reference source not found.**).~~ If this bit is set to '1', CDW11.PC is set to '1'; and the address pointer specifies Controller Memory Buffer, then, the address pointer in a Create I/O Submission Queue command (refer to **Error! Reference source not found.**) or a Create I/O Completion Queue command (refer to **Error! Reference source not found.**) shall be Dword aligned. If this bit is cleared to '0', then the I/O Submission Queues and I/O Completion Queues contained in the Controller Memory Buffer are aligned as defined by the PRP1 field of a Create I/O Submission Queue command (refer to **Error! Reference source not found.**) or a Create I/O Completion Queue command (refer to **Error! Reference source not found.**).
07	RO	Impl Spec	**CMB Data Metadata Mixed Memory Support (CDMMMS):** If this bit is set to '1', then the restriction on data and metadata use of Controller Buffer Memory by a command as defined in section 0 is not enforced. If this bit is cleared to '0', then the restriction on data and metadata use of Controller Buffer Memory by a command as defined in section 0 is enforced.
06	RO	Impl Spec	**CMB Data Pointer and Command Independent Locations Support (CDPCILS):** If this bit is set to '1', then the restriction that the PRP Lists and SGLs shall not be located in the Controller Buffer Memory if the command that they are associated with is not located in the Controller Buffer Memory is not enforced (refer to section 0). If this bit is cleared to '0', then that restriction is enforced.
05	RO	Impl Spec	**CMB Data Pointer Mixed Locations Support (CDPMLS):** If this bit is set to '1', then the restriction that for a particular PRP List or SGL associated with a single command, all memory that is associated with that particular PRP List or SGL shall reside in either the Controller Memory Buffer or outside the Controller Memory Buffer, is not enforced (refer to section 0). If this bit is cleared to '0', then that restriction is enforced.
04	RO	Impl Spec	**CMB Queue Physically Discontiguous Support (CQPDS):** If this bit is set to '1', then the restriction that for all queues in the Controller Memory Buffer, the queue shall be physically contiguous, is not enforced (refer to section 0). If this bit is cleared to '0', then that restriction is enforced.
03	RO	Impl Spec	**CMB Queue Mixed Memory Support (CQMMS):** If this bit is set to '1', then for a particular queue placed in the Controller Memory Buffer, the restriction that all memory associated with that queue shall reside in the Controller Memory Buffer is not enforced (refer to section 0). If this bit is cleared to '0', then that requirement is enforced.
02:00	RO	Impl Spec	**Base Indicator Register (BIR):** Indicates the Base Address Register (BAR) that contains the Controller Memory Buffer. For a 64-bit BAR, the BAR for the lower 32-bits of the address is specified. Values 000b, 010b, 011b, 100b, and 101b are valid. The address specified by the BAR shall be 4 KiB aligned.

Modify a portion of Figure 104 in section 4.2 as shown below:

4.2 Submission Queue Entry – Command Format

…

Figure 5: Command Dword 0

Bits	Description
31:16	**Command Identifier (CID):** This field specifies a unique identifier for the command when combined with the Submission Queue identifier. The value of FFFFh should not be used as the Error Information log page (refer to section 5.14.1.1) uses this value to indicate an error is not associated with a particular command.
15:14	**PRP or SGL for Data Transfer (PSDT):** This field specifies whether PRPs or SGLs are used for any data transfer associated with the command. PRPs shall be used for all Admin commands for NVMe over PCIe implementations. SGLs shall be used for all Admin and I/O commands for NVMe over Fabrics implementations. This field shall be set to 01b for NVMe over Fabrics revision 1.0 implementations. The definition is described in the table below. **Value** / **Definition** 00b: PRPs are used for this transfer. 01b: SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of a single contiguous physical buffer that is byte aligned. 10b: SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of an SGL segment containing exactly one SGL Descriptor that is qword aligned. 11b: Reserved If there is metadata that is not interleaved with the logical block data, as specified in the Format NVM command, then the Metadata Pointer (MPTR) field is used to point to the metadata. The definition of the Metadata Pointer field is dependent on the setting in this field. Refer to **Error! Reference source not found.**.
13:10	Reserved
09:08	**Fused Operation (FUSE):** In a fused operation, a complex command is created by "fusing" together two simpler commands. Refer to section **Error! Reference source not found.**. This field specifies whether this command is part of a fused operation and if so, which command it is in the sequence. **Value** / **Definition** 00b: Normal operation 01b: Fused operation, first command 10b: Fused operation, second command 11b: Reserved
07:00	**Opcode (OPC):** This field specifies the opcode of the command to be executed.

Modify a portion of section 4.3 as shown below:

4.3 Physical Region Page Entry and List

…

A physical region page list (PRP List) is a set of PRP entries in a single page of contiguous memory. A PRP List describes additional PRP entries that could not be described within the command itself. Any PRP entries described within the command are not duplicated in a PRP List. If the amount of data to transfer requires multiple PRP List memory pages, then the last PRP entry before the end of the memory page shall be a pointer to the next PRP List, indicating the next segment of the PRP List. Figure 109 shows the layout of a PRP List where each PRP entry identifies memory pages that are physically contiguous. Figure TBD shows the layout of a PRP List where each PRP entry identifies a different memory page (i.e., the memory pages are not physically contiguous).

Figure 109: PRP List Layout for Physically Contiguous Memory Pages

63 ... *n*+1	*n* ... 0
Page Base Address ~~*k*~~*p*	0h

Figure 109: PRP List Layout for Physically Contiguous Memory Pages

63 *n*+1	*n* 0
Page Base Address ~~*k*~~*p*+1	0h
…	
Page Base Address ~~*k*~~*p*+~~*m*~~*q*	0h
Page Base Address ~~*k*~~*p*+~~*m*~~*q* +1	0h

Figure TBD: PRP List Layout for Physically Non-Contiguous Memory Pages

63 *n*+1	*n* 0
Page Base Address ~~*k*~~*p*	0h
Page Base Address ~~*m*~~*q*	0h
…	
Page Base Address ~~*n*~~*r*	0h
Page Base Address ~~*p*~~*s*	0h

Modify a portion of section 4.5 (Metadata Region as shown below:

4.5 Metadata Region (MR)

…
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 and alignment of the Metadata Region is indicated by the Metadata Pointer within the command. ~~The Metadata Pointer within the command shall be dword aligned.~~

…

Modify a portion of Figure 126 in section 4.6.1.2.1 as shown below:

4.6.1.2.1 Generic Command Status Definition

…

Figure 6: Status Code – Generic Command Status Values

Value	Description
...	
0Dh	**Invalid SGL Segment Descriptor:** The command includes an invalid SGL Last Segment or SGL Segment descriptor. This may occur when the SGL segment pointed to by an SGL Last Segment descriptor contains an SGL Segment descriptor or an SGL Last Segment descriptor~~ or an SGL Segment descriptor~~. This may occur when an SGL Last Segment descriptor contains an invalid length (i.e., a length of 0h or 1h that is not a multiple of 16).

Modify a portion of section 4.7 as shown below:

4.7 Controller Memory Buffer

…

The contents of the Controller Memory Buffer are ~~initially~~ undefined as the result of:

- the CMBMSC.CMSE bit transitioning from ‘0’ to ‘1’;

- a Controller Reset; or
- a Function Level Reset.

Host software should initialize any memory in the Controller Memory Buffer before being referenced (e.g., a Completion Queue shall be initialized by host software in order for the Phase Tag to be used correctly).

Modify a portion of section 4.8 (Persistent Memory Region) as shown below:

4.8 Persistent Memory Region

…

The PMR write elasticity buffer size along with the PMR sustained write throughput allows a host to determine the amount of time for a read associated with a persistent memory region write barrier mechanism to complete.

Support for PRPs, SGL Lists, Completion Queues, and Submission Queues in the Persistent Memory Region is outside the scope of this specification. If the host attempts to use the Persistent Memory Region for a PRP, SGL List, Completion Queue, or Submission Queue, the controller may abort the command with a status code of Invalid Field in Command.

Modify a portion of Figure 151 (Create I/O Completion Queue – Command Dword 11) in section 5.3 as shown below:

5.3 Create I/O Completion Queue command

…

Figure 7: 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 this bit is cleared to '0' and CAP.CQR is set to '1', then 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 fail the command with Invalid Use of Controller Memory Buffer status.

Modify a portion of Figure 155 (Create I/O Submission Queue – Command Dword 11) in section 5.4 as shown below:

5.4 Create I/O Submission Queue command

…

Figure 8: Create I/O Submission Queue – Command Dword 11

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:16</td><td>Completion Queue Identifier (CQID): This field indicates the identifier of the I/O Completion Queue to utilize for any command completions entries associated with this Submission Queue.

If the value specified:

a) is 0h (i.e., the Admin Completion Queue), then the controller should return an error of Invalid Queue Identifier;

b) is outside the range supported by the controller, then the controller should return an error of Invalid Queue Identifier; or

c) is within the range supported by the controller and does not identify an I/O Completion Queue that has been created, then the controller should return an error of Completion Queue Invalid.</td></tr>
<tr><td>15:03</td><td>Reserved</td></tr>
<tr><td>02:01</td><td>Queue Priority (QPRIO): This field indicates the priority class to use for commands within this Submission Queue. This field is only used when the weighted round robin with urgent priority class is the arbitration mechanism selected, the field is ignored if weighted round robin with urgent priority class is not used. Refer to section Error! Reference source not found..

<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Urgent</td></tr><tr><td>01b</td><td>High</td></tr><tr><td>10b</td><td>Medium</td></tr><tr><td>11b</td><td>Low</td></tr></table></td></tr>
<tr><td>00</td><td>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’, then 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 fail the command with Invalid Use of Controller Memory Buffer status.</td></tr>
</table>

Modify portions of section 5.9.1 as shown below:

5.9.1 Command Completion

When the command is completed, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command. Command specific status values that may be returned are dependent on the Directive Type, refer to section **Error! Reference source not found.**.

Directive Receive command specific status values are defined in Figure TBD.

Figure TBD: Directive Receive – Command Specific Status Values

Value	Description
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).

Modify a portion of Figure 180 in section 5.12.1 as shown below:

5.12.1 Command Completion

…

Figure 9: Firmware Image Download – Command Specific Status Values

Value	Description
14h	**Overlapping Range:** This error is indicated if the firmware image has overlapping ranges. This error may ~~is~~ indicate~~d if~~ 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.

Modify a portion of section 5.13 (Get Features command) as shown below:

5.13 Get Features command

...

The Get Features command uses the Data Pointer, Command Dword 10, and Command Dword ~~11~~14~~,~~ fields. ~~and~~ The use of Command Dword ~~14~~11 field~~s~~ is Feature specific. If not used by a Feature, then Command Dword 11 is reserved unless otherwise stated. All other command specific fields are reserved.

Modify a portion of Figure 193 in section 5.14.1.1 as shown below:

5.14.1 Log Specific Information

…

Figure 10: Get Log Page – Log Page Identifiers

Log Identifier	Scope	Log Page Name	Reference Section
…			
02h	~~NVM subsystem~~Controller [1] Namespace [2]	SMART / Health Information	0
…			

Modify a portion of Figure 193 in section 5.14.1.1 as shown below:

5.14.1.1 Error Information (Log Identifier 01h)

…

Figure 11: Get Log Page – Error Information Log Entry (Log Identifier 01h)

Bytes	Description
…	
13:12	~~**Status Field:** This field indicates the Status Field for the command that completed. The Status Field is located in bits 15:01, bit 0 corresponds to the Phase Tag posted for the command.. If the error is not specific to a particular command, then this field reports the most applicable status value.~~ **Bits** / **Description** 15:~~0~~1 — **Status Field:** This field indicates the Status Field for the command that completed. If the error is not specific to a particular command, then this field reports the most applicable status value. 0 — **Phase Tag:** This field may indicate the Phase Tag posted for the command.

Modify a portion of Figure 194 in section 5.14.1.2 as shown below:

5.14.1.2 SMART / Health Information (Log Identifier 02h)

…

Figure 12: Get Log Page – SMART / Health Information Log

Bytes	Description
…	
63:48	**Data Units Written:** Contains the number of 512 byte data units the host has written to the controller; this value does not include metadata. This value is reported in thousands (i.e., a value of 1 corresponds to 1,000 units of 512 bytes written) and is rounded up (e.g., one indicates that the number of 512 byte data units written is from 1 to 1,000, three indicates that the number of 512 byte data units written is from 2,001 to 3,000). When the LBA size is a value other than 512 bytes, the controller shall convert the amount of data written to 512 byte units. For the NVM command set, logical blocks written as part of Write operations shall be included in this value. Write Uncorrectable commands and Write Zeroes commands shall not impact this value. A value of 0h in this field indicates that the number of Data Units Written is not reported.

Modify a portion of Figure 217 in section 5.14.1.13.1.4 as shown below:

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

…

Figure 13: Power-on or Reset Event (Event Type 04h)

Bytes	Description
07:00	**Firmware Revision:** Contains the firmware revision that becomes effective when CC.EN transitions from ‘0’ to ‘1’.

Figure 13: Power-on or Reset Event (Event Type 04h)

Bytes	Description
EL-VSIL-1:08	**Reset Information List:** Contains a list of one or more Controller Reset Information descriptors (refer to **Error! Reference source not found.**). If virtualization management is not implemented, then the list shall contain a Controller Reset Information descriptor for every controller in the NVM subsystem. If virtualization management is implemented, then the list shall contain a Controller Reset Information descriptor for every primary controller. The Controller Reset Information descriptor is shown in **Error! Reference source not found.**. EL is the value from the Event Length field in the Persistent Event Log event header (refer to Figure 212) and VSIL is the value from the Vendor Specific Information Length field in the Persistent Event Log header.

Modify a portion of Figure 219 in section 5.14.1.13.1.5 as shown below:

5.14.1.13.1.5 NVM Subsystem Hardware Error Event (Event Type 05h)

…

Figure 14: NVM Subsystem Hardware Error Event Format (Event Type 05h)

Bytes	Description
01:00	**NVM Subsystem Hardware Error Event Code:** This field contains a code (refer to **Error! Reference source not found.**) indicating the type of NVM subsystem hardware error that is being reported.
03:02	Reserved
M+3:04~~03~~	**Additional Hardware Error Information:** This field contains additional information about the hardware error event indicated in the NVM Subsystem Hardware Error Event Code field (refer to **Error! Reference source not found.**). Where M is the number of bytes of additional hardware error information. This field is omitted if the subsystem hardware error being reported does not contain additional hardware error information (i.e., if the number of bytes of additional hardware error information, M, is 0h).

Modify a portion of Figure 222 in section 5.14.1.13.1.6 as shown below:

5.14.1.13.1.6 Change Namespace Event (Event Type 06h)

…

Figure 15: Change Namespace Event Data Format (Event Type 06h)

Bytes	Value
…	
15:08	**Namespace Size (NSZE):** For a create operation, contains the NSZE value from the Identify Namespace data structure in the Namespace Management command (refer to **Error! Reference source not found.**). For a delete operation that specifies a single namespace this field contains the value from the NSZE field of the Identify Namespace data (refer to Figure 19) for the namespace being deleted. For a delete operation that specifies all namespaces this field is reserved.
23:16	Reserved

Figure 15: Change Namespace Event Data Format (Event Type 06h)

Bytes	Value
31:24	**Namespace Capacity (NCAP):** For a creation operation, contains the NCAP value from the Identify Namespace data structure in the Namespace Management command (refer to **Error! Reference source not found.**). For a delete operation that specifies a single namespace this field contains the value from the NCAP field of the Identify Namespace data (refer to Figure 19) for the namespace being deleted. For a delete operation that specifies all namespaces this field is reserved.
…	

Modify a portion of Figure 224 in section 5.14.1.13.1.8 as shown below:

5.14.1.13.1.8 Format NVM Completion Event (Event Type 08h)

…

Figure 16: 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 lowest numerical value that was available for reporting in the FPI field of 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 Fformat NVM command completed, refer to Figure 19) during the format operation. For a Format NVM command that formats all namespaces this field shall be cleared to 0h.
05	**Format NVM Status:** Contains the status of the format operation. **Bits** / **Definition** 7:2 – Reserved 1 – **Incomplete Format:** If set to '1', then the format operation modified one or more LBAs but did not complete successfully. If set to '1', then the Format NVM Error bit shall be set to '1'. If cleared to '0' then then the format operation either did not modify any LBAs or the format operation completed successfully. 0 – **Format NVM Error:** If set to '1', then the format operation did not complete successfully. If cleared to '0', then the format operation completed successfully.
07:06	**Completion Information:** Contains a vendor specific value that may provide more information about the completion of the format operation (e.g., if the format operation did not complete successfully, then this field may contain a vendor specific code that indicates a vendor specific reason).
11:08	**Status Field:** Contains the value that was reported in the status code field for the completion queue entry, if any, for the Format NVM command associated with this event. If no completion queue entry was reported, then this field shall be cleared to 0h.

Modify portions of section 5.14.1.13.1.11 as shown below:

5.14.1.13.1.11 Set Feature Event (Event Type 0Bh)

The Set Feature Event persists the data of a successful Set Features command. The event contains a Persistent Event Log Event header and the Set Feature Event data (refer to Figure 227).

The Set Feature Event shall set the Persistent Event Log Event Format Header:

- Event Type field to 0Bh; and
- Event Type Revision Field to 01h.

A Set Feature Event shall be recorded in the Persistent Event Log when the following criteria are met:

a) A Set Features command completes successfully;
b) The Feature Identifier in that Set Features command is supported to be logged in the Persistent Event Log; and
c) There is a change to the controller settings for the Feature Identifier in that Set Features command (i.e., the same setting is not set again).

A Set Feature Event may be recorded in the Persistent Event Log when there is no change to the controller settings for the Feature Identifier in that Set Features command if the following criteria are met:

a) A Set Features command completes successfully; and
b) The Feature Identifier in that Set Features command is supported to be logged in the Persistent Event Log.

The Feature Identifiers that may be supported to be logged in the Persistent Event Log are shown in Figure 422, Figure 423, Figure 430, and Figure 431.

The Command Dwords and Memory Buffer logged in the Set Feature Event data use the same formats as the formats defined by the Set Features and Get Features commands.

Figure 227: Set Feature Event Data Format

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>03:00</td><td>Set Feature Event Layout: Defines the number of Command Dwords and the amount of data in the Memory Buffer from the Set Features command associated with this event.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>31:16</td><td>Memory Buffer Count: Defines the number of bytes from the memory buffer that are logged in the Memory Buffer field. A value of 0h indicates that the Memory Buffer field does not exist.</td></tr>
<tr><td>15:04</td><td>Reserved</td></tr>
<tr><td>03</td><td>Logged Command Completion Dword 0: If set to ‘1’, then Dword 0 of the command completion for the Set Features command is included in the log. If cleared to ‘0’, then Dword 0 of the command completion command for the Set Features command is not included in the log.</td></tr>
<tr><td>02:00</td><td>Dword Count: contains the number of consecutive Dwords starting with Dword 10 from the Set Features command that are reported in the Command Dwords field. The values 0h and 7h are reserved.</td></tr>
</table></td></tr>
<tr><td>(Dword Count * 4)+3: 4</td><td>Command Dwords: Contains a sequential list of Command Dwords from the Set Features command starting with Command Dword 10. The number of entries in the list is specified by the Command Dword Count field. All non-reserved Command Dwords specified by the Set Features command for the Feature Identifier shall be logged. The Command Dwords are ordered as defined by command format for the Admin Command Set and NVM Command in Figure 105.</td></tr>
<tr><td>Data Buffer Count + (Dword Count * 4)+4: (Dword Count * 4)+4</td><td>Memory Buffer: Contains the data in the memory buffer for the Set Features command.

If the Memory Buffer Count field is cleared to a value on 0h, then this field does not exist in the logged event.</td></tr>
<tr><td>Data Buffer Count + (Dword Count * 4)+8: Data Buffer Count + (Dword Count * 4)+5</td><td>Command Completion Dword 0: If the Logged Command Completion Dword 0 bit is set to ‘1’, then this field contains the Dword 0 value from the Set Features command completion. If the Logged Command Completion Dword 0 bit is cleared to ‘0’, then this field is not logged.</td></tr>
</table>

Modify a portion of Figure 230 and Figure 231 in section 5.14.1.13.1.14 as shown below:

5.14.1.13.1.14 Vendor Specific Event (Event Type DEh)

…

Figure 17: Vendor Specific Event Format (Event Type DEh)

Bytes	Description
M-1:0	**Vendor Specific Event Descriptor 0:** Contains the first vendor specific event descriptor (refer to Figure 18). Where M is the length of this vendor specific event descriptor.
…	
EL-VSIL-1: EL-VSIL-K	**Vendor Specific Event Descriptor N:** Contains the last vendor specific event descriptor (refer to Figure 18). Where EL is the value from the Event Length field in the Persistent Event Log event header (refer to Figure 212), VSIL is the value from the Vendor Specific Information Length field in the Persistent Event Log header, and K is the length of this vendor specific event descriptor (refer to Figure 231).

…

Figure 18: Vendor Specific Event Descriptor Format

Bytes	Description
01:00	**Vendor Specific Event Code:** Contains a vendor specific code that uniquely identifies the type of event that is described in the data that follows. All vendor specific events of the same type should report the same Vendor Specific Event Code field value.
02	**Vendor Specific Event Data Type:** Contains a code indicating the type of data reported in the Vendor Specific Event Data field (refer to **Error! Reference source not found.**).
03	**UUID Index:** UUID Index (refer to **Error! Reference source not found.**) at the time of this event for the vendor that defined this event.
05:04	**Vendor Specific Event Data Length (VSEDL)**: Contains the length in bytes of the Vendor Specific Event Data field.
Vendor Specific Event Data, if any (i.e., VSEDL > 0)	
VSEDL~~M~~+5:06	**Vendor Specific Event Data:** Contains vendor specific data that is associated with this event and is of the type specified in the Vendor Specific Event Data Type field. ~~Where M is the length of the vendor specific event data.~~

Modify a portion of Figure 245 in section 5.15.2.1 as shown below:

5.15.2.1 Identify Namespace data structure (CNS 00h)

Figure 19: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M [1]	Description
...		
15:08	M	**Namespace Capacity (NCAP):** This field indicates the maximum number of logical blocks that may be allocated in the namespace at any point in time. The number of logical blocks is based on the formatted LBA size. ~~This field is undefined prior to the namespace being formatted. This field is used in the case of thin provisioning and reports a value that is smaller than or equal to the Namespace Size.~~ Spare LBAs are not reported as part of this field. ~~A logical block is allocated when it is written with a Write or Write Uncorrectable command. A logical block may be deallocated using the Dataset Management, Sanitize, or Write Zeroes command.~~ Refer to section 6.1.TBD for details on the usage of this field.
23:16	M	**Namespace Utilization (NUSE):** This field indicates the current number of logical blocks allocated in the namespace. This field is smaller than or equal to the Namespace Capacity. The number of logical blocks is based on the formatted LBA size. ~~When using the NVM command set: A logical block is allocated when it is written with a Write or Write Uncorrectable command. A logical block may be deallocated using the Dataset Management, Sanitize, or Write Zeroes command.~~ ~~A controller may report NUSE equal to NCAP at all times if the product is not targeted for thin provisioning environments.~~ ~~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 **Error! Reference source not found.**) or the ANA Persistent Loss state (refer to section **Error! Reference source not found.**), then this field shall be cleared to 0h.~~ Refer to section 6.1.TBD for details on the usage of this field.

Figure 19: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M [1]	Description
24	M	**Namespace Features (NSFEAT):** This field defines features of the namespace. Bits 7:5 are reserved. Bit 4 (**OPTPERF**) if set to '1': • indicates that the fields NPWG, NPWA, NPDG, NPDA, and NOWS are defined for this namespace and should be used by the host for I/O optimization (refer to section 8.25); and • NOWS defined for this namespace shall adhere to Optimal Write Size field setting defined in NVM Sets Attributes Entry (refer to **Error! Reference source not found.**) for the NVM Set with which this namespace is associated. If cleared to '0', then: • the controller does not support the fields NPWG, NPWA, NPDG, NPDA, and NOWS for this namespace; and • Optimal Write Size field in NVM Sets Attributes Entry (refer to **Error! Reference source not found.**) for the NVM Set with which this namespace is associated should be used by the host for I/O optimization. Bit 3 (**UIDREUSE**) if set to '1' indicates that the value in the NGUID field for this namespace, if non-zero, is never reused by the controller and that the value in the EUI64 field for this namespace, if non-zero, is never reused by the controller. If cleared to '0', then the NGUID value may be reused and the EUI64 value may be reused by the controller for a new namespace created after this namespace is deleted. This bit shall be cleared to '0' if both NGUID and EUI64 fields are cleared to 0h. Refer to section 0. Bit 2 (**DAE**) if set to '1' indicates that the controller supports the Deallocated or Unwritten Logical Block error for this namespace. If cleared to '0', then the controller does not support the Deallocated or Unwritten Logical Block error for this namespace. Refer to section 0. Bit 1 (**NSABP**) if set to '1' indicates that the fields NAWUN, NAWUPF, and NACWU are defined for this namespace and should be used by the host for this namespace instead of the AWUN, AWUPF, and ACWU fields in the Identify Controller data structure. If cleared to '0', then the controller does not support the fields NAWUN, NAWUPF, and NACWU for this namespace. In this case, the host should use the AWUN, AWUPF, and ACWU fields defined in the Identify Controller data structure in **Figure 20**. Refer to section **Error! Reference source not found.**. Bit 0 (**THINP**) if set to '1' indicates that the namespace supports thin provisioning. ~~Specifically, the Namespace Capacity reported may be less than the Namespace Size. When this feature is supported and the Dataset Management command is supported, then deallocating LBAs shall be reflected in the Namespace Utilization field. Bit 0 i~~If cleared to '0' indicates that thin provisioning is not supported ~~and the Namespace Size and Namespace Capacity fields report the same value~~. Refer to section 6.1.TBD for details on the usage of this bit.
35:34	O	**Namespace Atomic Write Unit Normal (NAWUN):** This field indicates the namespace specific size of the write operation guaranteed to be written atomically to the NVM during normal operation. If the NSABP bit is cleared to '0', then this field is reserved. A value of 0h indicates that the size for this namespace is the same size as that reported in the AWUN field of the Identify Controller data structure. All other values specify a size in terms of logical blocks using the same encoding as the AWUN field. Refer to section **Error! Reference source not found.**.

Figure 19: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M [1]	Description
37:36	O	**Namespace Atomic Write Unit Power Fail (NAWUPF):** This field indicates the namespace specific size of the write operation guaranteed to be written atomically to the NVM during a power fail or error condition. If the NSABP bit is cleared to '0', then this field is reserved. A value of 0h indicates that the size for this namespace is the same size as that reported in the AWUPF field of the Identify Controller data structure. All other values specify a size in terms of logical blocks using the same encoding as the AWUPF field. Refer to section **Error! Reference source not found.**.
39:38	O	**Namespace Atomic Compare & Write Unit (NACWU):** This field indicates the namespace specific size of the write operation guaranteed to be written atomically to the NVM for a Compare and Write fused command. If the NSABP bit is cleared to '0', then this field is reserved. A value of 0h indicates that the size for this namespace is the same size as that reported in the ACWU field of the Identify Controller data structure. All other values specify a size in terms of logical blocks using the same encoding as the ACWU field. Refer to section **Error! Reference source not found.**.
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 OPTPEFT bit is cleared to '0', then this field is reserved. The size indicated should be less than or equal to Maximum Data Transfer Size (MDTS) 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 **Error! Reference source not found.** for how this field is utilized to improve performance and endurance.
67:66	O	**Namespace Preferred Write Alignment (NPWA):** This field indicates the recommended write alignment in logical blocks for this namespace. This is a 0's based value. If the OPTPEFT bit is cleared to '0', then this field is reserved. The value of this field may change if the namespace is reformatted. Refer to section **Error! Reference source not found.** for how this field is utilized to improve performance and endurance.
69:68	O	**Namespace Preferred Deallocate Granularity (NPDG):** This field indicates the recommended granularity in logical blocks for the Dataset Management command with the Attribute – Deallocate bit set to '1' in Dword 11.This is a 0's based value. If the OPTPEFT bit is cleared to '0', then this field is reserved. The value of this field may change if the namespace is reformatted. The size should be a multiple of Namespace Preferred Deallocate Alignment (NPDA). Refer to section **Error! Reference source not found.** for how this field is utilized to improve performance and endurance.
71:70	O	**Namespace Preferred Deallocate Alignment (NPDA):** This field indicates the recommended alignment in logical blocks for the Dataset Management command with the Attribute – Deallocate bit set to '1' in Dword 11.This is a 0's based value. If the OPTPEFT bit is cleared to '0', then this field is reserved. The value of this field may change if the namespace is reformatted. Refer to section **Error! Reference source not found.** for how this field is utilized to improve performance and endurance.

Figure 19: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M [1]	Description
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 OPTPEFT bit is cleared to '0', then this field is reserved. The size indicated should be less than or equal to Maximum Data Transfer Size (MDTS) 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). Refer to section **Error! Reference source not found.** for how this field is utilized to improve performance and endurance.

Modify a portion of Figure 247 in section 5.15.2.2 as shown below:

5.15.2.2 Identify Controller data structure (CNS 01h)

…

Figure 20: Identify – Identify Controller Data Structure

Bytes	O/M [1]	Description
Controller Capabilities and Features		
…		
77	M	**Maximum Data Transfer Size (MDTS):** This field indicates the maximum data transfer size for a command that transfers data between host-accessible memory (refer to section 1.6.TBD)~~accessible by the host (e.g., host memory, Controller Memory Buffer (refer to section 0))~~ and the controller. The host should not submit a command that exceeds this maximum data transfer size. If a command is submitted that exceeds this transfer size, then the command is aborted with a status of Invalid Field in Command. The value is in units of the minimum memory page size (CAP.MPSMIN) and is reported as a power of two (2^*n*). A value of 0h indicates that there is no maximum data transfer size. This field includes the length of metadata, if metadata is interleaved with the logical block data. This field does not apply to commands that do not transfer data between host-accessible memory ~~accessible by the host~~ and the controller (e.g., the Verify command, the Write Uncorrectable command, and the Write Zeroes command); there is no maximum data transfer size for those commands. If SGL Bit Bucket descriptors are supported, their lengths shall be included in determining if a command exceeds the Maximum Data Transfer Size for destination data buffers. Their length in a source data buffer is not included for a Maximum Data Transfer Size calculation.
…		

Figure 20: Identify – Identify Controller Data Structure

Bytes	O/M [1]	Description
533:532	O	**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 **Error! Reference source not found.**. 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 fail the command with a status code of Atomic Write Unit Exceeded ~~Invalid Field in Command~~. If Compare and Write is not a supported fused command, then this field shall be 0h.

Modify portions of Figure 266 in section 5.20.1 as shown below:

5.20.1 Command Completion

…

Figure 21: 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 specification).
15h	**Namespace Insufficient Capacity:** Creating the namespace requires more unallocated capacity ~~free space~~ than is currently available (refer to Unallocated NVM Capacity field in Figure 247). The Command Specific Information field of the Error Information Log specifies the total amount of unallocated NVM capacity required to create the namespace in bytes.
16h	**Namespace Identifier Unavailable:** The number of namespaces supported has been exceeded.
1Bh	**Thin Provisioning Not Supported:** Thin provisioning is not supported by the controller.
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).
24h	**ANA Group Identifier Invalid:** The specified ANA Group Identifier (ANAGRPID) is not supported in the submitted command. This may be due to various conditions, including: a) specifying an ANAGRPID that does not exist; b) the controller does not allow an ANAGRPID to be specified (i.e., bit 7 in the ANACAP field is cleared to '0'); or c) the specified ANAGRPID is not supported by the controller processing the command (e.g., the specified value exceeds ANAGRPMAX (refer to Figure 20)). If the host specified a non-zero ANAGRPID, retrying the command with the ANAGRPID field cleared to 0h may succeed.

Modify a portion of section 5.21 as shown below:

5.21 Set Features command

...

The Set Features command uses the Data Pointer, Command Dword 10, and Command Dword ~~11~~14~~,~~. The use of Command Dword ~~12~~11, Command Dword ~~13~~12, Command Dword ~~14~~13, and Command Dword 15 fields is Feature specific. If Command Dword 11, Command Dword 12, Command Dword 13, or Command Dword 15 fields are not used, then the Command Dwords are reserved. ~~All other command specific fields are reserved.~~

Modify a portion of section 5.21.1 as shown below:

5.21.1 Feature Specific Information

…

If the controller does not support a changeable value for a Feature (e.g., the Feature is not changeable), 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.

Modify a portions of section 5.21.1.5 as shown below:

5.21.1.5 Error Recovery (Feature Identifier 05h)

...

Figure 22: Error Recovery – Command Dword 11

Bits	Description
31:17	Reserved
16	**Deallocated or Unwritten Logical Block Error Enable (DULBE):** If set to '1', then the Deallocated or Unwritten Logical Block error is enabled for the namespace specified in the NSID field. If cleared to '0', then the Deallocated or Unwritten Logical Block error is disabled for the namespace specified in the NSID field. Host software shall only enable this error if the DAE bit in the NSFEAT field ~~bit 2~~ is set to '1' in the Identify Namespace data structure. The default value for this bit shall be '0'. Refer to section 0.

Modify a portions of section 5.23 as shown below:

5.23 Format NVM command – NVM Command Set Specific

…

Figure 23: Format NVM – Operation Scope

FNA Bit [1]	NSID	Format Operation
0	FFFFFFFFh	All namespaces attached to the controller. Other namespaces are not affected.
0	Any allocated value (refer to section **Error! Reference source not found.**)	Particular namespace specified. Other namespaces are not affected.

Figure 23: Format NVM – Operation Scope

FNA Bit [1]	NSID	Format Operation
1	Any allocated value (refer to section **Error! Reference source not found.**) or FFFFFFFFh	All allocated namespaces in the NVM subsystem
NOTES: 1. For a Format NVM command with Secure Erase, this column refers to bit 1 in the FNA field in the Identify Controller data structure (refer to Figure 20) and bit 0 in the FNA field is ignored. For a Format NVM command without Secure Erase, this column refers to bit 0 in the FNA field, and bit 1 in the FNA field is ignored.		

…

~~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 specification). The Format NVM command may fail if there are outstanding I/O commands being processed by the controller 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.~~

The Format NVM command may be aborted with a status code defined in this specification under circumstances defined by a security specification (e.g., invalid security state as specified in TCG Storage Interface Interactions specification). If there are I/O commands being processed for a namespace, then a Format NVM command that is submitted affecting that namespace may be aborted; if aborted, then a status code of Command Sequence Error should be returned. If a Format NVM command is in progress, then an I/O command that is submitted for any namespace affected by that Format NVM command may be aborted; if aborted, then a status code of Format in Progress should be returned.

...

Modify a portion of Figure 329 in section 5.23.1 as shown below:

5.23.1 Command Completion

…

Figure 24: 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; or 3) the specified format is not available in the current configuration.~~; or~~ ~~4) invalid security state (refer to TCG Storage Interface Interactions specification), etc.~~
0Ch	**Command Sequence Error:** The command was aborted due to a protocol violation in a multi-command sequence.
15h	**Operation Denied:** The command was denied due to lack of access rights. Refer to the appropriate security specification.
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).
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 specification).

Modify a portion of Figure 332 in section 5.24.1 as shown below:

5.24.1 Command Completion

…

Figure 25: Sanitize – Command Specific Status Values

Value	Description
10h	**Firmware Activation Requires NVM Subsystem Reset:** The sanitize operation could not be started because a firmware activation is pending.
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19)
23h	**Sanitize Prohibited While Persistent Memory Region is Enabled:** A sanitize operation is prohibited while the Persistent Memory Region is enabled.

Modify a portion of section 6.1.6 as shown below:

6.1.6 NSID and Namespace Usage

…

UIDREUSE b~~B~~it ~~3~~ in the NSFEAT field (refer to **Figure 19**) indicates NGUID and EUI64 reuse characteristics.

~~The Namespace Size (NSZE) field in the Identify Namespace data structure defines the total size of the namespace in logical blocks (LBA 0 through *n-1*). The Namespace Utilization (NUSE) field in the Identify Namespace data structure defines the number of logical blocks currently allocated in the namespace. The Namespace Capacity (NCAP) field in the Identify data structure defines the maximum number of logical blocks that may be allocated at one time as part of the namespace in a thin provisioning usage model. The following relationship holds: Namespace Size >= Namespace Capacity >= Namespace Utilization.~~

~~If the controller supports Asymmetric Namespace Access Reporting (i.e., bit 3 set to '1' in the CMIC field in the Identify Controller data structure (refer to Figure 20)), then the NUSE field (refer to Figure 19) and the NVMCAP field (refer to Figure 19) are cleared to 0h if the relationship between the controller and the namespace is in the ANA Inaccessible state (refer to section **Error! Reference source not found.**) or the ANA Persistent Loss state (refer to section **Error! Reference source not found.**). The Namespace Attribute Changed event is not generated for changes to these fields that result from ANA state changes as described in **Error! Reference source not found.**. The host uses the Asymmetric Namespace Access Change Notices as an indication of these changes.~~

A namespace may or may not have a relationship to a Submission Queue; this relationship is determined by the host software implementation. The controller shall support access to any active namespace from any I/O Submission Queue.

Add the following section after section 6.1.6 as shown below:

6.1.TBD Namespace Size, Capacity and Utilization

The Identify Namespace data structure (refer to Figure 245) contains related fields reporting the Namespace Size, Capacity and Utilization:

- The Namespace Size (NSZE) field defines the total size of the namespace in logical blocks (LBA 0 through n-1).
- The Namespace Capacity (NCAP) field defines the maximum number of logical blocks that may be allocated at any point in time.
- The Namespace Utilization (NUSE) field defines the number of logical blocks currently allocated in the namespace.

The following relationship holds: Namespace Size >= Namespace Capacity >= Namespace Utilization.

When the THINP bit in the NSFEAT field is set to ‘1’, the controller:

- may report a value in the Namespace Capacity field that is smaller than the value in the Namespace Size field; and
- shall track the number of allocated blocks in the Namespace Utilization field.

When the THINP bit in the NSFEAT field is cleared to ‘0’, the controller:

- shall report a value in the Namespace Capacity field that is equal to the Namespace Size; and
- may report a value in the Namespace Utilization field that is always equal to the value in the Namespace Capacity field.

A logical block shall be marked as allocated when it is written with:

- a Write command;
- a Write Uncorrectable command; or
- a Write Zeroes command that does not deallocate the logical block (refer to section 6.7.1.1).

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

- a Write command not addressing the logical block; or
- a Write Zeroes command not addressing the logical block (refer to section 6.7.1.1).

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

- a Dataset Management command (refer to section 6.7.1.1); or
- a Write Zeroes command addressing the logical block (refer to section 6.7.1.1); or
- a Sanitize operation.

If the controller supports Asymmetric Namespace Access Reporting (i.e., bit 3 set to ‘1’ in the CMIC field in the Identify Controller data structure (refer to Figure 247)), then the NUSE field (refer to Figure 245) and the NVMCAP field (refer to Figure 245) are cleared to 0h if 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). The Namespace Attribute Changed event is not generated for changes to these fields that result from ANA state changes as described in Figure 147. The host uses the Asymmetric Namespace Access Change Notices as an indication of these changes.

Modify a portion of Figure 355 in section 6.5 as shown below:

6.5 End-to-end Protection Information

…

Figure 26: Protection Information Field Definition

<table>
<tr><th>Description</th></tr>
<tr><td>Protection Information Action (PRACT): The protection information action bit indicates the action to take for the protection information. ~~This bit is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this bit is ignored. Refer to section Error! Reference source not found..
<table>
<tr><th>PRACT Value</th><th>Metadata Size</th><th>Description</th></tr>
<tr><td>1b</td><td>8 Bytes</td><td>The protection information is stripped (read) or inserted (write).</td></tr>
<tr><td>1b</td><td>> 8 Bytes</td><td>The protection information is passed (read) or replaces the first or last 8 bytes of the metadata (write).</td></tr>
<tr><td>0b</td><td>any</td><td>The protection information is passed (read and write).</td></tr>
</table>
</td></tr>
<tr><td>Protection Information Check (PRCHK): The protection information check field specifies the fields that shall be checked as part of end-to-end data protection processing. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section Error! Reference source not found..
<table>
<tr><th>Bit</th><th>Definition</th></tr>
<tr><td>02</td><td>If set to ‘1’ enables protection information checking of the Guard field. If cleared to ‘0’, the Guard field is not checked.</td></tr>
<tr><td>01</td><td>If set to ‘1’ enables protection information checking of the Application Tag field. If cleared to ‘0’, the Application Tag field is not checked.</td></tr>
<tr><td>00</td><td>If set to ‘1’ enables protection information checking of the Logical Block Reference Tag field. If cleared to ‘0’, the Logical Block Reference Tag field is not checked.</td></tr>
</table>
</td></tr>
</table>

Modify a portion of section 6.6 as shown below:

6.6 Compare command

…

Figure 360: Compare – Command Dword 14

Bits	Description
31:00	**Expected Initial Logical Block Reference Tag (EILBRT):** This field specifies the Initial Logical Block Reference Tag expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Figure 361: Compare – Command Dword 15

Bits	Description
31:16	**Expected Logical Block Application Tag Mask (ELBATM):** This field specifies the Application Tag Mask expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.
15:00	**Expected Logical Block Application Tag (ELBAT):** This field specifies the Application Tag expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Modify a portion of Figure 365 in section 6.7 as shown below:

6.7 Dataset Management command

…

Figure 27: Dataset Management – Command Dword 11

Bit	Description
31:03	Reserved
02	**Attribute – Deallocate (AD):** If set to '1', then the NVM subsystem may deallocate all provided ranges. The data returned for ~~a~~ logical blocks that were deallocated ~~range~~ is specified in section 0. The data and metadata for logical blocks that are not deallocated by the NVM subsystem are not changed as the result of a Dataset Management command.
01	**Attribute – Integral Dataset for Write (IDW):** If set to '1', then the dataset should be optimized for write access as an integral unit. The host expects to perform operations on all ranges provided as an integral unit for writes, indicating that if a portion of the dataset is written it is expected that all of the ranges in the dataset are going to be written.
00	**Attribute – Integral Dataset for Read (IDR):** If set to '1', then the dataset should be optimized for read access as an integral unit. The host expects to perform operations on all ranges provided as an integral unit for reads, indicating that if a portion of the dataset is read it is expected that all of the ranges in the dataset are going to be read.

Modify a portion of section 6.7.1.1 as shown below:

6.7.1.1 Deallocated or Unwritten Logical Blocks

A logical block that has never been written to, or which has been deallocated using the Dataset Management command, the Write Zeroes command or the Sanitize command is called a deallocated or unwritten logical block ~~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~~.

Using the Error Recovery feature (refer to section 5.21.1.5), host software may select the behavior of the controller when reading deallocated or unwritten blocks. The controller shall fail Read, Verify, or Compare commands that include deallocated or unwritten blocks with a status of Deallocated or Unwritten Logical Block if that error has been enabled using the DULBE bit in the Error Recovery feature. If the Deallocated or Unwritten Logical error is not enabled, the values read from a deallocated or unwritten block and its metadata (excluding protection information) shall be:

- all bytes cleared to 0h if bits 2:0 in the DLFEAT field are set to 001b;
- all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 010b; or
- either all bytes cleared to 0h or all bytes set to FFh if bits 2:0 in the DLFEAT field are set to 000b.

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. A deallocated or unwritten block is no longer deallocated or unwritten when the logical block is written. Read operations and Verify operations do not affect the deallocation status of a 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), or 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 19) may report the values read from a deallocated logical block and its metadata.~~

…

Modify a portion of section 6.7.2 as shown below:

6.7.2 Command Completion

…

Figure 28: Dataset Management – Command Specific Status Values

Value	Description
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).
80h	**Conflicting Attributes:** The attributes specified in the command are conflicting.
82h	**Attempted Write to Read Only Range:** The controller may optionally report this status if a Deallocate is attempted for a read only range. The controller shall not return this status value 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 **Error! Reference source not found.**).

Modify a portion of section 6.9 as shown below:

6.9 Read command

…

Figure 374: Read – Command Dword 14

Bits	Description
31:00	**Expected Initial Logical Block Reference Tag (EILBRT):** This field specifies the Initial Logical Block Reference Tag expected value. ~~This field only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Figure 375: Read – Command Dword 15

Bits	Description
31:16	**Expected Logical Block Application Tag Mask (ELBATM):** This field specifies the Application Tag Mask expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.
15:00	**Expected Logical Block Application Tag (ELBAT):** This field specifies the Application Tag expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Modify a portion of section 6.14 as shown below:

6.14 Verify command

…

Figure 396: Verify – Command Dword 14

Bits	Description
31:00	**Expected Initial Logical Block Reference Tag (EILBRT):** This field specifies the Initial Logical Block Reference Tag expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Figure 396: Verify – Command Dword 14

Bits	Description

Figure 397: Verify – Command Dword 15

Bits	Description
31:16	**Expected Logical Block Application Tag Mask (ELBATM):** This field specifies the Application Tag Mask expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.
15:00	**Expected Logical Block Application Tag (ELBAT):** This field specifies the Application Tag expected value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Modify a portion of section 6.15 as shown below:

6.15 Write command

…

Figure 404: Write – Command Dword 14

Bits	Description
31:00	**Initial Logical Block Reference Tag (ILBRT):** This field specifies the Initial Logical Block Reference Tag value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Figure 405: Write – Command Dword 15

Bits	Description
31:16	**Logical Block Application Tag Mask (LBATM):** This field specifies the Application Tag Mask value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.
15:00	**Logical Block Application Tag (LBAT):** This field specifies the Application Tag value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Modify a portion of Figure 406 in section 6.15.1 as shown below:

6.15.1 Command Completion

…

Figure 29: Write – Command Specific Status Values

Value	Description
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).
80h	**Conflicting Attributes:** The attributes specified in the command are conflicting.
81h	**Invalid Protection Information:** The Protection Information Field (PRINFO) (refer to **Error! Reference source not found.**) settings specified in the command are invalid for the Protection Information with which the namespace was formatted (refer to the PI field in **Error! Reference source not found.** and the DPS field in Figure 19) or the ILBRT field is invalid (refer to section 0).

Figure 29: Write – Command Specific Status Values

Value	Description
82h	**Attempted Write to Read Only Range:** The LBA range specified contains read-only blocks. The controller shall not return this status value 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 **Error! Reference source not found.**).

Modify a portion of Figure 409 in section 6.16.1 as shown below:

6.16.1 Command Completion

…

Figure 30: Write Uncorrectable – Command Specific Status Values

Bit	Description
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).
82h	**Attempted Write to Read Only Range:** The LBA range specified contains read-only blocks. The controller shall not return this status value 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 **Error! Reference source not found.**).

Modify a portion of section 6.17 as shown below:

6.17 Write Zeroes command

The Write Zeroes command is used to set a range of logical blocks to zero. Non-PI related metadata for this command, if any, shall be all bytes cleared to 0h. The protection information for logical blocks written to the media is updated based on CDW12.PRINFO. If the Protection Information Action bit (PRACT) is cleared to '0', then the protection information for this command shall be all zeroes. If the Protection Information Action bit (PRACT) is set to '1', then the protection information shall be based on the End-to-end Data Protection Type Settings (DPS) field in the Identify Namespace data structure (refer to Figure 19) and the CDW145~~5~~.~~E~~ILBRT, CDW15.~~E~~LBATM, and CDW15.~~E~~LBAT fields in the Write Zeroes command.

…

Figure 412: Write Zeroes – Command Dword 14

Bits	Description
31:00	**Initial Logical Block Reference Tag (ILBRT):** This field indicates the Initial Logical Block Reference Tag value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Figure 413: Write Zeroes – Command Dword 15

Bits	Description
31:16	**Logical Block Application Tag Mask (LBATM):** This field indicates the Application Tag Mask value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.
15:00	**Logical Block Application Tag (LBAT):** This field indicates the Application Tag value. ~~This field is only used if the namespace is formatted to use end-to-end protection information.~~ If the namespace is not formatted to use end-to-end protection information, then this field is ignored. Refer to section **Error! Reference source not found.**.

Modify a portion of Figure 414 in section 6.17.1 as shown below:

6.17.1 Command Completion

…

Figure 31: Write Zeroes – Command Specific Status Values

Value	Description
20h	**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.19).
81h	**Invalid Protection Information:** The Protection Information Field (PRINFO) (refer to **Error! Reference source not found.**) settings specified in the command are invalid for the Protection Information with which the namespace was formatted (refer to the PI field in **Error! Reference source not found.** and the DPS field in Figure 19) or the ILBRT field is invalid (refer to section 0).
82h	**Attempted Write to Read Only Range:** The LBA range specified contains read-only blocks. The controller shall not return this status value 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 **Error! Reference source not found.**).

Modify a portion of Figure 418 (I/O Controller – NVM Command Set Specific Admin Command Support) in section 7.1.1 as shown below:

7.1.1 I/O Controller

…

Figure 32: I/O Controller – NVM Command Set Specific Admin Command Support

Command	Command Support Requirements [1]
Format NVM	O
~~Security Send~~	~~O~~
Security Send	O
Security Receive	O
Sanitize	O
Get LBA Status	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited	

Modify a portion of section 7.1.2 as shown below:

7.1.2 Administrative Controller

An administrative controller is required to support the mandatory Admin commands listed in **Error! Reference source not found.**. An administrative controller may support one or more I/O Command Sets. When an Administrative controller supports an I/O Command Set, then only I/O Command Set Specific Admin commands may be supported since an Administrative controller only has an Admin Q~~q~~ueue and no I/O Q~~q~~ueues.

…

Modify a portion of section 7.3.2 as shown below (:

7.3.2 Controller Level Reset

The~~re are~~ following ~~five~~ methods ~~to~~ initiate a Controller Level Reset:

- NVM Subsystem Reset;

- Conventional Reset (refer to the PCI Express Base Specification ~~i.e., PCI Express Hot, Warm, or Cold reset~~);
- ~~PCI Express transaction layer Data Link Down status;~~
- Function Level Reset (refer to the PCI Express Base Specification ~~i.e., PCI reset~~); and
- Controller Reset (i.e., CC.EN transitions from '1' to '0').

…

In all Controller Level Reset cases except a Controller Reset, the PCI register space is reset as defined by the PCI Express Base specification. Refer to the PCI Express specification for further details.

To continue after a Controller Level Reset, the host shall:

Modify a portion of section 7.4.2 as shown below:

7.4.2 Queue Coordination

There is one Admin Q~~q~~ueue pair associated with multiple I/O queue pairs. The Admin Submission Queue and Completion Queue are used to carry out functions that impact the entire controller. An I/O Submission Queue and Completion Queue may be used to carry out I/O (read/write) operations and may be distributed across CPU cores and threads.

…

Modify a portion of section 7.9 as shown below:

7.9 NVMe Qualified Names

…

The reverse domain name in an NQN that uses this format shall not be "org.nvmexpress".

The following are examples of NVMe Qualified Names that may be generated by "Example NVMe, Inc."

- nqn.2014-08.com.example:nvme:nvm-subsystem-sn-d78432; and
- nqn.2014-08.com.example:nvme.host.sys.xyz.

…

Modify a portion of section 7.11 as shown below:

7.11 Unique Identifier

...

If the UIDREUSE bit ~~3~~ in the NSFEAT field is cleared to '0', then a controller may reuse a non-zero NGUID/EUI64 value for a new namespace after the original namespace using the value has been deleted. If the UIDREUSE bit ~~3 in NSFEAT~~ is set to '1', then a controller shall not reuse a non-zero NGUID/EUI64 for a new namespace after the original namespace using the value has been deleted.

Modify a portion of section 8.3.1.5 as shown below:

8.3.1.5 Control of Protection Information Checking – PRCHK

…

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

Modify a portion of section 8.12 as shown below:

8.12 Namespace Management (Optional)

The Namespace Management capability consist of the Namespace Management command (refer to section 5.20) and the Namespace Attachment command (refer to section 5.19). The Namespace Management command is used to create a namespace or delete a namespace. The Namespace Attachment command is used to attach and detach controllers from a namespace. The Namespace Management capability is intended for use during manufacturing or by a system administrator.

…

Modify a portion of section 8.13 as shown below:

Boot Partitions provide an optional area of NVM storage that may be read by the host without the host initializing queues or enabling the controller. The simplified interface to access Boot Partitions may be used for platform initialization code (e.g., a bootloader that is executed from host ROM) to boot to a pre-OS environment (e.g., UEFI) instead of storing the image on another storage medium (e.g., SPI flash). Refer to section **Error! Reference source not found.** for the procedure to read the contents of a Boot Partition.

…

Modify Portions of section 8.17.1 (Configuring and Managing Events) as shown below:

8.17.1 Configuring and Managing Events

…

The host configures events using a Set Features command for each Endurance Group.

…