LEGAL NOTICE:

This erratum to the NVM Express revision 1.3a specification 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 to the NVM Express revision 1.3 specification subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **"© 2007 - 2016 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

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

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

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

NVM Express™ Technical Errata

Errata ID	**004a**
Revision Date	**03/22/2018**
Affected Spec Ver.	**NVM Express™ 1.3a**
Corrected Spec Ver.	**Replaces ECN-004**

Errata Author(s)

Name	Company
Fred Knight	NetApp Inc.
Tom Friend	Toshiba
Michael Allison	SK Hynix Inc
Judy Brock	Samsung
Christoph Hellwig, Naren Nadesan	WDC
Jonathan Hughes	Intel
Eric Peterson	Synopsys
David Black, Kevin Marks	Dell EMC
Peter Onufryk	Microsemi

Errata Overview

Clarifications on the definition of “0’s based” values.
Make “Privileged Actions” referenceable.
Update references section (and update many uses of references)
Clarify use of NSID FFFFFFFFh in various admin commands
Clarification that it is not optional to post CQEs on command completion
Clarification of Log Page scope
Clarification of Self-test Result “Segment number”
Clarification of some RPMB field descriptions
Clarification of reset behavior related to VQ resources and VI resources
Clarification for systems using UUID only
Clarification of LBA Range feature
Clarifications of HMB use
Clarifications of Format NVM command
Clarification of definition of RTD3 Resume Latency
Clarification of Reservation Register action
Clarification of VWC (Volatile Write Cache)

Revision History

Revision Date	Change Description
08/30/2017	Initial creation
10/12/2017	Updates based on weekly con-call
10/16/2017	Updates to RTD3 and NSID = FFFFFFFFh
10/23/17	Move power items to ECN-005
10/26/17	Update Flush Command and associated text (push Flush to ECN-005)
11/03/2017	Updates to 0's based language, LBA Ranges, and Unique IDs.
11/07/2017	Add TP4000 to list of affected specs
11/09/2017	Move TP4000 changes into TP4000a
11/30/2017	Editorial items from con-call.
1/9/2018	Correct change list (to remove those that got moved to ECN-005), update copyright notice to 2018.
2/15/2018	Address member review comments (move FNA text to ECN-005 and add "unless otherwise specified" in Figure 42 footnote 3).
3/22/2018	Correct errors found after ECN-004 publication
3/22/2018	Ratified

Incompatible Changes

The LBA Range feature was silent on results if multiple Set Features were processed; this is now described as subsequent commands replacing the settings of any earlier command.

The Number of Queues feature required it be issued before queue creation, but did not specify the error if the host violated this requirement. The error is now specified.

The HMB feature was silent on actions associated with multiple attempts to enable the HMB. This is now specified as an error. It was also silent on checks necessary during disabling HMB; it is now specified that other inputs are ignored during a disable operation.

The command description section (Figure 11) seemed to imply that the NSID could be set to FFFFFFFFh for any command, even for Read commands and Write commands; even though that was not intended. The specification now lists the Admin commands for which an NSID of FFFFFFFFh apply.

Description of Specification Changes

Modify a portion of 1.5 (Conventions) as shown below:

1.5 Conventions

…

When a register field is referred to in the document, the convention used is “Register Symbol.Field Symbol”. For example, the PCI command register parity error response enable field is referred to by the name CMD.PEE. If the register field is an array of bits, the field is referred to as “Register Symbol.Field Symbol (array offset to element)”.

A ~~0-based~~ 0’s based value is a numbering scheme ~~for~~ in which the number 0h ~~actually corresponds to~~ represents a value of 1h and thus produces the pattern of 0h ~~=~~ represents 1h, 1h ~~=~~ represents 2h, 2h ~~=~~ represents 3h, etc. In this numbering scheme, there is no~~t a~~ method ~~for specifying~~ to represent the value of 0h. Values in this specification are 1-based (i.e., the number 1h ~~corresponds to~~ represents a value of 1h, 2h~~=~~ represents 2h, etc.) unless otherwise specified.

When a size is stated in the document as KB, the convention used is 1KB = 1024 bytes.

…

Delete section 1.6.26 (privileged actions) as shown below:

~~**1.6.26 privileged actions**~~

~~An action (command, register write, etc.) that affects or has the potential to affect the state of the entire NVM subsystem and not only the controller and/or namespace with which the action is associated. Admin commands that are privileged include Namespace Management, Namespace Attachment, Virtualization Management, Format NVM, and Sanitize. A privileged register action is NVM subsystem reset. Vendor specific commands and registers may also be privileged.~~

Modify a portion of section 1.9 (References) as shown below:

1.9 References

…

JEDEC JESD218B-01: Solid State Drive (SSD) Requirements and Endurance Test Method standard. Available from http://www.jedec.org.

…

PCI Local Bus S~~s~~pecification, R~~r~~evision 3.0. Available from http://www.pcisig.com.

PCI Express Base S~~s~~pecification, ~~revision 2.1~~ Revision 3.1a. Available from http://www.pcisig.com.

PCI Bus Power Management Interface S~~s~~pecification Revision 1.2. Available from http://www.pcisig.com.

…

PCI Firmware ~~3.0~~ S~~s~~pecification Revision 3.2. Available from http://www.pcisig.com.

PCI Code and ID Assignment Specification Revision 1.9, 18 May, 2017. Available from http://www.pcisig.com.

RFC 4122, P. Leach, M. Mealling, and R. Salz, “A Universally Unique Identifier (UUID) URN Namespace”, July 2005. Available from https://www.ietf.org/rfc.html.

~~RFC 4301, S. Kent, S. and K. Seo, “Security Architecture for the Internet Protocol”, December 2005. Available from https://www.ietf.org/rfc.html.~~

RFC 6234, D. Eastlake 3rd~~, D.~~ and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", May 2011. Available from https://www.ietf.org/rfc.html.

~~UEFI 2.3.1 specification. Available from http://www.uefi.org.~~

UEFI Specification Version 2.7A, September 2017. Available from http://www.uefi.org.

~~Trusted Computing Group~~TCG Storage Architecture Core ~~s~~Specification, Version 2.01 Revision 1.00. Available from http://www.trustedcomputinggroup.org.

~~Trusted Computing Group~~TCG Storage Interface Interactions Specification ~~(SIIS)~~, Version 1.05 Revision 1.00. Available from http://www.trustedcomputinggroup.org.

Modify a portion of section 2.1.5 (Offset 09h: CC - Class Code) as shown below:

2.1.5 Offset 09h: CC - Class Code

Fields in the Class Code register are described in the PCI Code and ID Assignment Specification.

Bits	Type	Reset	Description
23:16	RO	01h	**Base Class Code (BCC):** Indicates the base class code as a mass storage controller.
15:08	RO	08h	**Sub Class Code (SCC):** Indicates the sub class code as a Non-Volatile Memory controller.
07:00	RO	02h	**Programming Interface (PI):** This field specifies the programming interface of the controller is NVM Express. ~~(Note: The PCI SIG documentation refers to this as Enterprise NVMHCI.)~~

Modify a portion of section 4.2 (Submission Queue Entry - Command Format) as shown below:

4.2 Submission Queue Entry - Command Format

…

Figure 11: Command Format – Admin and NVM Command Set

Bytes	Description
03:00	**Command Dword 0 (CDW0):** This field is common to all commands and is defined in Figure 10.
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. ~~Setting this value to FFFFFFFFh causes the command to be applied to all namespaces attached to this controller, unless otherwise specified.~~ The value FFFFFFFFh in this field is a broadcast value (refer to section 6.1), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 41 and Figure 42 for Admin commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 6.1.4) ~~ID~~ in a command that uses the namespace identifier ~~ID~~ shall cause the controller to abort the command with status Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 6.1.2) ~~ID~~ in a command that uses the namespace identifier ~~ID~~ shall cause the controller to abort the command with status Invalid Namespace or Format, unless otherwise specified.
…	

Modify a portion of section 4.6.1 (Status Field Definition) as shown below:

4.6.1 Status Field Definition

…

Figure 29: Completion Queue Entry: Status Field

Bit	Description
31	**Do Not Retry (DNR):** If set to '1', indicates that if the same command is re-submitted ~~it~~ to any controller in the NVM subsystem then that re-submitted command is expected to fail. If cleared to '0', indicates that the same command may succeed if retried. If a command is aborted due to time limited error recovery (refer to section 5.21.1.5, this field should be cleared to '0'. If the SCT and SC fields are cleared to 0h then this field should be cleared to '0'.
…	

Modify a portion of section 4.6.1.2.1 (Generic Command Status Definition) as shown below:

4.6.1.2.1 Generic Command Status Definition

Completion queue entries with a Status Code type of Generic Command Status indicate a status value associated with the command that is generic across many different types of commands.

Figure 31: Status Code – Generic Command Status Values

Value	Description
00h	**Successful Completion:** The command completed ~~successfully~~ without error.
01h	**Invalid Command Opcode:** ~~The associated command opcode field is not valid.~~ A reserved coded value or an unsupported value in the command opcode field.
02h	**Invalid Field in Command:** A reserved coded value or an unsupported value in a defined field (other than the opcode field). The field may be in the command parameters as part of the Submission Queue Entry or in data structures pointed to by the command parameters.
…	
06h	**Internal Error:** The command was not completed successfully due to an internal error. Details on the internal device error are ~~returned~~ available to report as an asynchronous event. Refer to ~~section 5.2~~ Figure 47 for Internal Error Asynchronous Event Information.
…	
0Ch	**Command Sequence Error:** The command was aborted due to a protocol violation in a multi-command sequence (e.g. a violation of the Security Send and Security Receive sequencing rules in the TCG Storage Synchronous Interface Communications protocol (refer to TCG Storage Architecture Core Specification)).
…	
15h	**Operation Denied:** The command was denied due to lack of access rights. Refer to the appropriate security specification (e.g., TCG ~~SIIS~~ Storage Interface Interactions Specification). For media access commands, the Access Denied status code should be used instead.
…	

Modify a portion of section 4.6.1.2.3 (Media and Data Integrity Errors Definition) as shown below:

4.6.1.2.3 Media and Data Integrity Errors Definition

Completion queue entries with a Status Code Type of Media and Data Integrity Errors indicate an error associated with the command that is due to an error associated with the NVM media or a data integrity type error.

Figure 36: Status Code – Media and Data Integrity Error Values, NVM Command Set

Value	Description
...	
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 ~~SIIS~~ Storage Interface Interactions Specification).
...	

Modify a portion of section 5 (Admin Command Set) as shown below:

5 Admin Command Set

The Admin Command Set defines the commands that may be submitted to the Admin Submission Queue.

...

Figure 41: Opcodes for Admin Commands

Opcode by Field			Combined Opcode[2]	O/M[1]	Namespace Identifier Used[3]	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[4]				
0b	000 00b	00b	00h	M	No	Delete I/O Submission Queue
0b	000 00b	01b	01h	M	No	Create I/O Submission Queue
0b	000 00b	10b	02h	M	Yes	Get Log Page
0b	000 01b	00b	04h	M	No	Delete I/O Completion Queue
0b	000 01b	01b	05h	M	No	Create I/O Completion Queue
0b	000 01b	10b	06h	M	Yes	Identify
0b	000 10b	00b	08h	M	No	Abort
0b	000 10b	01b	09h	M	Yes	Set Features
0b	000 10b	10b	0Ah	M	Yes	Get Features
0b	000 11b	00b	0Ch	M	No	Asynchronous Event Request
0b	000 11b	01b	0Dh	O	Yes	Namespace Management
0b	001 00b	00b	10h	O	No	Firmware Commit
0b	001 00b	01b	11h	O	No	Firmware Image Download
0b	001 01b	00b	14h	O	Yes	Device Self-test
0b	001 01b	01b	15h	O	Yes[6]	Namespace Attachment
0b	001 10b	00b	18h	NOTE 5	No	Keep Alive
0b	001 10b	01b	19h	O	Yes[6]	Directive Send
0b	001 10b	10b	1Ah	O	Yes[6]	Directive Receive
0b	001 11b	00b	1Ch	O	No	Virtualization Management
0b	001 11b	01b	1Dh	O	No	NVMe-MI Send
0b	001 11b	10b	1Eh	O	No	NVMe-MI Receive
0b	111 11b	00b	7Ch	O	No	Doorbell Buffer Config
0b	111 11b	11b	7Fh	O	Refer to the NVMe over Fabrics specification.	
	I/O Command Set Specific					
1b	na	NOTE 4	80h – BFh	O		I/O Command Set specific

Opcode by Field			Combined Opcode[2]	O/M[1]	Namespace Identifier Used[3]	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[4]				
	Vendor Specific					
1b	na	NOTE 4	C0h – FFh	O		Vendor specific

NOTES:

1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. A subset of commands uses the Namespace Identifier field (CDW1.NSID). If the Namespace Identifier field is used, then the value FFFFFFFFh is supported in this field unless footnote 6 in this figure indicates that a specific command does not support that value. When this field is not used, the field shall be cleared to 0h.
4. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
5. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory.
6. This command does not support the use of the Namespace Identifier field (CDW1.NSID) set to FFFFFFFFh.

Figure 42 defines Admin commands that are specific to the NVM Command Set.

Figure 42: Opcodes for Admin Commands – NVM Command Set Specific

Opcode (07)	Opcode (06:02)	Opcode (01:00)	Opcode[2]	O/M[1]	Namespace Identifier Used[3]	Command
Generic Command	Function	Data Transfer[4]				
1b	000 00b	00b	80h	O	Yes	Format NVM
1b	000 00b	01b	81h	O	NOTE 5	Security Send
1b	000 00b	10b	82h	O	NOTE 5	Security Receive
1b	000 01b	00b	84h	O	No	Sanitize

NOTES:

1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. A subset of commands uses the Namespace Identifier field (CDW1.NSID). If the Namespace Identifier field is used, then unless otherwise specified, the value FFFFFFFFh is supported in this field. When this field is not used, the field shall be cleared to 0h.
4. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
5. The use of the Namespace Identifier is Security Protocol specific.

Modify a portion of section 5.5 and 5.5.1 (Delete I/O Completion Queue command) as shown below:

5.5 Delete I/O Completion Queue command

…

5.5.1 Command Completion

~~A completion queue entry is posted to the Admin Completion Queue when the indicated I/O Completion Queue has been deleted.~~ Upon completion of the Delete I/O command, the controller posts a completion queue entry to the Admin Completion Queue. Delete I/O Completion Queue command specific status values are defined in Figure 60.

Modify a portion of section 5.11 (Firmware Commit command) as shown below (changes shown here are based on changes published in ECN-002):

5.11 Firmware Commit command

…

Figure 76: Firmware Commit – Command Dword 10

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31</td><td>Boot Partition ID (BPID): Specifies the Boot Partition that shall be used for the Commit Action, if applicable.</td></tr>
<tr><td>30:06</td><td>Reserved</td></tr>
<tr><td>05:03</td><td>Commit Action (CA): This field specifies the action that is taken (refer to section 8.1) on the image downloaded with the Firmware Image Download command or on a previously downloaded and placed image. The actions are indicated in the following table.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Downloaded image replaces the image specified by the Firmware Slot field. This image is not activated.</td></tr>
<tr><td>001b</td><td>Downloaded image replaces the image specified by the Firmware Slot field. This image is activated at the next reset.</td></tr>
<tr><td>010b</td><td>The image specified by the Firmware Slot field is activated at the next reset.</td></tr>
<tr><td>011b</td><td>The image specified by the Firmware Slot field is requested to be activated immediately without reset.</td></tr>
<tr><td>100-101b</td><td>Reserved</td></tr>
<tr><td>110b</td><td>Downloaded image replaces the Boot Partition specified by the Boot Partition ID field.</td></tr>
<tr><td>111b</td><td>Mark the Boot Partition specified in the BPID field as active and update BPINFO.ABPID.</td></tr>
</table></td></tr>
<tr><td>02:00</td><td>Firmware Slot (FS): Specifies the firmware slot that shall be used for the Commit Action, if applicable. If the value specified is 0h, then the controller shall choose the firmware slot (slot 1 – 7) to use for the operation.</td></tr>
</table>

5.11.1 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.

Requests that specify activation of a new firmware image at the next reset and return with status code value of 00h, any Controller Level Reset defined in section 7.3.2 activates the specified firmware.

…

Modify a portion of section 5.12.1 (Firmware Image Download command) as shown below:

5.12 Firmware Image Download command

…

5.12.1 Command Completion

~~A completion queue entry is posted to the Admin Completion Queue if this portion of the firmware image has been received by the controller.~~ Upon completion of the Firmware Image Download command, the controller

posts a completion queue entry to the Admin Completion Queue. Firmware Image Download command specific status values are defined in Figure 81.

Modify a portion of section 5.13.2 (Get Features command) as shown below:

5.13 Get Features command

…

5.13.2 Command Completion

~~A completion queue entry is posted to the Admin Completion Queue if the controller has completed returning any attributes associated with the Feature.~~ Upon completion of the Get Features command, the controller posts a completion queue entry to the Admin Completion Queue. Depending on the Feature Identifier, Dword 0 of the completion queue entry may contain feature information (refer to section 5.21.1).

Modify a portion of section 5.14 (Get Log Page command) as shown below:

5.14 Get Log Page command

…

Figure 88: Get Log Page – Command Dword 12

Bit	Description
31:00	**Log Page Offset Lower (LPOL):** The log page offset specifies the location within a log page to start returning data from. This field specifies the lower 32 bits of the log page offset. ~~This field is Dword aligned such that the lower two bits shall be~~ The offset shall be Dword aligned, indicated by bits 1:0 being cleared to 00b. The controller is not required to check that bits 1:0 are cleared to 00b. The controller may report an error of Invalid Field in Command if bits 1:0 are not cleared to 00b. If the controller does not report an error of Invalid Field in Command, then the controller shall operate as if bits 1:0 are cleared to 00b.

Modify a portion of 5.14.1 (Log Specific Information) as shown below:

5.14.1 Log Specific Information

Figure 90 and Figure 91 define the Log pages that may be retrieved with the Get Log Page command and the scope of the information that is returned in those Log pages.

Log pages that indicate a scope of NVM subsystem return information that is global to the NVM subsystem. Log pages that indicate a scope of controller return information that is specific to the controller that is processing the command. Log pages that indicate a scope of Namespace return information that is specific to the specified namespace. For log pages that indicate multiple scopes, the namespace identifier that is specified determines which information is returned. The definition of any individual field within a Log page may indicate a different scope that is specific to that individual field.

For Log Pages with a scope of NVM subsystem or controller (as shown in Figure 90 and Figure 91), the controller should abort commands that specify namespace identifiers other than 0h or FFFFFFFFh with status Invalid Field in Command. Otherwise the rules for namespace identifier usage in Figure 11 apply.

Figure 90: Get Log Page – Log Page Identifiers

Log Identifier	O/M	Scope	Description	Reference Section
00h	Reserved			
01h	M	Controller	Error Information	5.14.1.1
02h	M	Controller[1]	SMART / Health Information	5.14.1.2
	O	Namespace[2]		
03h	M	NVM subsystem	Firmware Slot Information	5.14.1.3
04h	O	Controller	Changed Namespace List	5.14.1.4
05h	O	Controller	Commands Supported and Effects	5.14.1.5
06h	O	NVM subsystem	Device Self-test	5.14.1.6
07h	O	Controller	Telemetry Host-Initiated	5.14.1.7
08h	O	Controller	Telemetry Controller-Initiated	5.14.1.8
09h – 6Fh	Reserved			
70h	Discovery (refer to the NVMe over Fabrics specification)			
71h – 7Fh	Reserved for NVMe over Fabrics			
80h – BFh	I/O Command Set Specific			
C0h – FFh	Vendor specific			
KEY: O/M definition: O = Optional, M = Mandatory				
NOTES: 1. For namespace identifiers of 0h or FFFFFFFFh 2. For namespace identifiers other than 0h or FFFFFFFFh				

~~O/M: O = Optional, M = Mandatory~~

Figure 91: Get Log Page – Log Page Identifiers, NVM Command Set Specific

Log Identifier	O/M	Scope	Description	Reference Section
80h	O	Controller	Reservation Notification	5.14.1.9.1
81h	O	NVM subsystem	Sanitize Status	5.14.1.9.2
82h – BFh	Reserved			
KEY: O/M definition: O = Optional, M = Mandatory				

~~O/M: O = Optional, M = Mandatory~~

Modify a portion of section 5.14.1.2 (Smart / Health Information) as shown below (changes shown here are based on changes published in ECN-002 and ECN-003):

5.14.1.2 SMART / Health Information (Log Identifier 02h)

This log page is used to provide SMART and general health information. The information provided is over the life of the controller and is retained across power cycles. ~~The log page shall be supported on a global basis.~~ To request the ~~global~~ controller log page, the namespace identifier specified is FFFFFFFFh. The ~~log page~~ controller may also ~~be supported~~ support requesting the log page on a per namespace basis, as indicated by bit 0 of the LPA field in the Identify Controller data structure in Figure 109.

If the log page is not supported on a per namespace basis, specifying a~~ny~~ namespace identifier other than 0h or FFFFFFFFh should abort the command with status Invalid Field in Command. If the controller does

not abort the command, then implementations compliant with this revision and earlier revisions of this specification return the controller log page. There is not namespace specific information defined in the SMART / Health log page in this revision of the specification, ~~thus~~ therefore the ~~global~~ controller log page and namespaces specific log page contain identical information.

…

Modify a portion of section 5.14.1.3 (Firmware Slot Information) as shown below:

5.14.1.3 Firmware Slot Information (Log Identifier 03h)

This log page is used to describe the firmware revision stored in each firmware slot supported. The firmware revision is indicated as an ASCII string. The log page also indicates the active slot number. The log page returned is defined in Figure 95. ~~This log page is global to the controller.~~

Modify a portion of section 5.14.1.6 (Device Self-test) as shown below:

5.14.1.6 Device Self-test (Log Identifier 06h)

…

Figure 99: Get Log Page - Self-test Result Data Structure

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>0</td><td>
Device Self-test Status: This field indicates the device self-test code and the status of the operation.

Bits 7:4 indicates the Self-test Code value that was specified in the Device Self-test command that started the device self-test operation that this Self-test Result Data Structure describes.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0h</td><td>Reserved</td></tr>
<tr><td>1h</td><td>Short device self-test operation</td></tr>
<tr><td>2h</td><td>Extended device self-test operation</td></tr>
<tr><td>3h – Dh</td><td>Reserved</td></tr>
<tr><td>Eh</td><td>Vendor specific</td></tr>
<tr><td>Fh</td><td>Reserved</td></tr>
</table>
Bits 3:0 indicates the result of the device self-test operation that this Self-test Result Data Structure describes.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>0h</td><td>Operation completed without error</td></tr>
<tr><td>1h</td><td>Operation was aborted by a Device Self-test command</td></tr>
<tr><td>2h</td><td>Operation was aborted by a Controller Level Reset</td></tr>
<tr><td>3h</td><td>Operation was aborted due to a removal of a namespace from the namespace inventory</td></tr>
<tr><td>4h</td><td>Operation was aborted due to the processing of a Format NVM command</td></tr>
<tr><td>5h</td><td>A fatal error or unknown test error occurred while the controller was executing the device self-test operation and the operation did not complete</td></tr>
<tr><td>6h</td><td>Operation completed with a segment that failed and the segment that failed is not known</td></tr>
<tr><td>7h</td><td>Operation completed with one or more failed segments and the first segment that failed is indicated in the Segment Number field</td></tr>
<tr><td>8h</td><td>Operation was aborted for unknown reason</td></tr>
<tr><td>9h – Eh</td><td>Reserved</td></tr>
<tr><td>Fh</td><td>Entry not used (does not contain a test result)</td></tr>
</table>
</td></tr>
<tr><td>1</td><td>Segment Number: This field indicates ~~which~~ the segment number (refer to section 8.11) where the first self-test failure occurred. ~~The field is ignored if the~~ If Device Self-test Status field bits [3:0] are ~~is~~ not set to 7h, then this field should be ignored.</td></tr>
<tr><td>...</td><td></td></tr>
</table>

Modify a portion of section 5.14.1.9.1 (Reservation Notification) as shown below:

5.14.1.9.1 Reservation Notification (Log Identifier 80h)

A Reservation Notification log page is created whenever an unmasked reservation notification occurs on any namespace that is attached to the controller. The Get Log Page command returns a data buffer containing a log page corresponding to a single reservation notification. The format of the log page is defined in Figure 103. ~~This log page is global to the controller.~~

Modify a portion of section 5.14.1.9.2 (Sanitize Status) as shown below:

5.14.1.9.2 Sanitize Status (Log Identifier 81h)

The Sanitize Status log page is used to report sanitize operation time estimates and information about the most recent sanitize operation (refer to section 8.15). The Get Log Page command returns a data buffer containing a log page formatted as defined in Figure 104. This log page ~~is global to the NVM subsystem and~~ shall be retained across power cycles and resets. This log page shall contain valid data whenever CSTS.RDY is set to '1'.

Modify a portion of section 5.14.2 (Get Log Page ->Command Completion) as shown below:

5.14.2 Command Completion

~~A completion queue entry is posted to the Admin Completion Queue if the log has been transferred to the memory buffer indicated in PRP Entry 1~~. Upon completion of the Get Log Page command, the controller posts a completion queue entry to the Admin Completion Queue. Get Log Page command specific status values are defined in Figure 105.

Modify a portion of section 5.15 and 5.15.1 (Identify command) as shown below (changes to Figure 106 are based on changes published in ECN-003):

5.15 Identify command

…

Figure 106: Identify – Data Structure Returned

CNS Value	O/M	Definition
…		
03h	M	A list of Namespace Identification Descriptor structures (refer to Figure 116) is returned to the host for the namespace specified in the Namespace Identifier (CDW1.NSID) field if it is an active NSID. The controller may return any number of variable length Namespace Identification Descriptor structures that fit into the 4096 byte Identify payload. All remaining bytes after the namespace identification descriptor structures should be cleared to 0h, and the host shall interpret a Namespace Identifier Descriptor Length (NIDL) value of 0h as the end of the list. If, while processing these descriptors, the host detects a descriptor type, that it does not recognize, then it should skip the unrecognized descriptor type and continue parsing the structure. A controller shall not return multiple descriptors with the same Namespace ~~Identification Descriptor~~ Identifier Type (NIDT). A controller shall return at least one descriptor identifying the namespace.
…		

…

Figure 109: Identify – Identify Controller Data Structure

<table>
<tr><th>Bytes</th><th>O/M</th><th>Description</th></tr>
<tr><td>...</td><td></td><td></td></tr>
<tr><td>87:84</td><td>M</td><td>RTD3 Resume Latency (RTD3R): This field indicates the ~~typical~~ expected latency in microseconds ~~resuming~~ to resume from Runtime D3 (RTD3). Refer to section 8.4.4 ~~for test conditions~~. A value of 0h indicates RTD3 Resume Latency is not reported.</td></tr>
<tr><td>...</td><td></td><td></td></tr>
<tr><td>315:312</td><td>O</td><td>Replay Protected Memory Block Support (RPMBS): This field indicates if the controller supports one or more Replay Protected Memory Blocks (RPMBs) and the capabilities. Refer to section 8.10.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:24</td><td>Access Size: If the Number of RPMB Units field is non-zero, then t~~T~~his field indicates the ~~size~~ number of 512B units of data that may be read or written per RPMB access by Security Send or Security Receive commands for this controller ~~in 512B units~~. This is a 0's based value. A value of 0h indicates ~~a size of~~ support for one unit of 512B of data.

If the Number of RPMB Units field is 0h, this field shall be ignored.</td></tr>
<tr><td>23:16</td><td>Total Size: If the Number of RPMB Units field is non-zero, then t~~T~~his field indicates the ~~size of~~ number of 128KB units of data in each RPMB supported in the controller ~~in 128KB units~~. This is a 0's based value. A value of 0h indicates ~~a size of~~ support for one unit of 128KB of data.

If the Number of RPMB Units field is 0h, this field shall be ignored.</td></tr>
<tr><td>15:06</td><td>Reserved</td></tr>
<tr><td>05:03</td><td>Authentication Method: This field indicates the authentication method used to access all RPMBs in the controller. The values for this field are:
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>HMAC SHA-256 (refer to RFC 6234)</td></tr>
<tr><td>001b-111b</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>02:00</td><td>Number of RPMB Units: This field indicates the number of RPMB targets the controller supports. All RPMB targets supported shall have the same capabilities as defined in the RPMBS field. A value of 0h indicates the controller does not support Replay Protected Memory Blocks. If this value is non-zero, then the controller shall support the Security Send and Security Receive commands.</td></tr>
</table></td></tr>
<tr><td>...</td><td></td><td></td></tr>
<tr><td>525</td><td>M</td><td>Volatile Write Cache (VWC): This field indicates attributes related to the presence of a volatile write cache in the ~~implementation~~controller.

Bits 7:1 are reserved.

Bit 0 if set to '1' indicates that a volatile write cache is present. If cleared to '0', a volatile write cache is not present.

If a volatile write cache is present, then the host ~~may issue Flush commands and~~ controls whether the volatile write cache is enabled with Set Features specifying the Volatile Write Cache feature identifier (refer to section 5.21.1.6). The Flush command (refer to section 6.8) is used to request that the contents of a volatile write cache be made non-volatile. ~~If a volatile write cache is not present, Flush commands complete successfully and have no effect, Set Features with the Volatile Write Cache identifier field set shall fail with Invalid Field status, and Get Features with the Volatile Write Cache identifier field set should fail with Invalid Field status.~~</td></tr>
<tr><td>...</td><td></td><td></td></tr>
</table>

Figure 110: Identify – Primary Controller Capabilities Structure

Bytes	Description
…	
41:40	**VQ Resources Flexible Allocated to Primary (VQRFAP):** This field indicates the total number of VQ Flexible Resources currently allocated to the primary controller. This value may change after a Controller Level Reset other than a Controller Reset (i.e., CC.EN transitions from ‘1’ to ‘0’) if a new value was set using the Virtualization Management command. The default value of this field is implementation specific.
…	
73:72	**VI Resources Flexible Allocated to Primary (VIRFAP):** This field indicates the total number of VI Flexible Resources currently allocated to the primary controller. This value may change after a Controller Level Reset other than a Controller Reset (i.e., CC.EN transitions from ‘1’ to ‘0’) if a new value was set using the Virtualization Management command. The default value of this field is implementation specific.
…	

Figure 114: Identify – Identify Namespace Data Structure, NVM Command Set Specific

<table>
<tr><th>Bytes</th><th>O/M</th><th>Description</th></tr>
<tr><td>…</td><td></td><td></td></tr>
<tr><td>33</td><td>O</td><td>Deallocate Logical Block Features (DLFEAT): This field indicates information about features that affect deallocating logical blocks for this namespace.

Bits 7:5 are reserved.

Bit 4 if set to ‘1’ indicates that the Guard field for deallocated logical blocks that contain protection information is set to the CRC for the value read from the deallocated logical block and its metadata (excluding protection information). If cleared to ‘0’ indicates that the Guard field for the deallocated logical blocks that contain protection information is set to FFFFh.

Bit 3 if set to ‘1’ indicates that the controller supports the Deallocate bit in the Write ~~Zeros~~ Zeroes command for this namespace. If cleared to ‘0’ indicates that the controller does not support the Deallocate bit in the Write ~~Zeros~~ Zeroes command for this namespace. This bit shall be set to the same value for all namespaces in the NVM subsystem.

Bits 2:0 indicate the values read from a deallocated logical block and its metadata (excluding protection information). The values for this field have the following meanings:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Not reported</td></tr>
<tr><td>001b</td><td>All bytes set to 00h</td></tr>
<tr><td>010b</td><td>All bytes set to FFh</td></tr>
<tr><td>011b – 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>…</td><td></td><td></td></tr>
<tr><td>119:104</td><td>O</td><td>Namespace Globally Unique Identifier (NGUID): This field contains a 128-bit value that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., controller reset, namespace format, etc.).

This field uses the EUI-64 based 16-byte designator format. Bytes 114:112 contain the 24-bit Organizationally Unique Identifier (OUI) value assigned by the IEEE Registration Authority. Bytes ~~NIDT~~119:115 contain an extension identifer assigned by the corresponding organization. Bytes 111:104 contain the vendor specific extension identifier assigned by the corresponding organization. See the IEEE EUI-64 guidelines for more information. This field is big endian (refer to section ~~7.10~~ 7.10.5).

The controller shall specify a globally unique namespace identifier in this field, ~~or~~ the EUI64 field, or a Namespace UUID in the Namespace Identification Descriptor when the namespace is created. If the controller is not able to ~~allocate~~ provide a globally unique identifier in this field, then this field shall be cleared to 0h. Refer to section 7.11.</td></tr>
<tr><td>127:120</td><td>O</td><td>IEEE Extended Unique Identifier (EUI64): This field contains a 64-bit IEEE Extended Unique Identifier (EUI-64) that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., controller reset, namespace format, etc.).

The EUI-64 is a concatenation of a 24-bit or 36-bit Organizationally Unique Identifier (OUI or OUI-36) value assigned by the IEEE Registration Authority and an extension identifier assigned by the corresponding organization. See the IEEE EUI-64 guidelines for more information. This field is big endian (refer to section ~~7.10~~ 7.10.4).

The controller shall specify a globally unique namespace identifier in this field, ~~or~~ the NGUID field, or a Namespace UUID in the Namespace Identification Descriptor when the namespace is created. If the controller is not able to ~~allocate~~ provide a globally unique 64-bit identifier in this field, then this field shall be cleared to 0h. Refer to section 7.11.</td></tr>
</table>

Bytes	O/M	Description
...		

...

Figure 116: Identify – Namespace Identification Descriptor

Bytes	Description
00h	**Namespace Identifier Type (NIDT):** This field indicates the data type contained in the Namespace Identifier field and the length for that type as defined in the following table. **Value** / **Length (NIDL)** / **Definition** 0h / / Reserved 1h / 8h / **IEEE Extended Unique Identifier:** The NID field contains a copy of the EUI64 field in the Identify Namespace data structure (refer to Figure 114). If the EUI64 field of the Identify Namespace data structure is not supported (i.e., EUI64 field is set to zero), the controller shall not report a Namespace Identification Descriptor with a value of type 1h. ~~For a Namespace Identifier Descriptor of type 1h the Namespace Identifier Length (NIDL) shall be set to 8h.~~ 2h / 10h / **Namespace Globally Unique Identifier:** The NID field contains a copy of the NGUID field in the Identify Namespace data structure (refer to Figure 114). If the NGUID field of the Identify Namespace data structure is not supported (i.e., the NGUID field is set to zero), the controller shall not report a Namespace Identification Descriptor with a value of type 2h. ~~For a Namespace Identifier Descriptor of type 2h the Namespace Identifier Length (NIDL) shall be set to 10h.~~ 3h / 10h / **Namespace UUID:** The NID field contains a 128-bit Universally Unique Identifier (UUID) as specified in RFC4122. Refer to section 7.10.6. If the namespace does not support an IEEE Extended Unique Identifier (i.e., EUI64 field is set to zero) and does not support a Namespace Globally Unique Identifier (i.e., the NGUID field is set to zero), then the namespace shall report a Namespace Identification Descriptor with a value of type 3h. ~~For a Namespace Identifier Descriptor of type 3h the Namespace Identifier Length (NIDL) shall be set to 10h.~~ 4h - FFh / / Reserved
~~0~~1~~h~~	**Namespace Identifier Length (NIDL):** This field contains the length in bytes of the Namespace Identifier (NID) field ~~below~~. The total length of the Namespace ~~Identifier~~ Identification Descriptor in bytes is the value in this field plus four. If this field is set to 0h it indicates the end of the Namespace Identifier Descriptor list.
~~0~~2~~h~~ – ~~0~~3~~h~~	Reserved
~~0~~4~~h~~ – (NIDL + 03h)	**Namespace Identifier (NID):** This field contains a value that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., controller reset, namespace format, etc.). The ~~exact~~ type of the value is specified by the Namespace Identifier Type (NIDT) field, and the size is specified by the Namespace Identifier Length (NIDL) field.

5.15.1 Command Completion

~~A completion queue entry is posted to the Admin Completion Queue if the Identify data structure has been transferred to the memory buffer indicated in PRP Entry 1~~. Upon completion of the Identify command, the controller posts a completion queue entry to the Admin Completion Queue.

Modify a portion of section 5.20.1 (Namespace Management command -> Command Completion) as shown below:

5.20 Namespace Management command

…

5.20.1 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.

Namespace Management command specific status values are defined in Figure 130.

Figure 130: 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, or 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 ~~SIIS~~ Storage Interface Interactions Specification)~~, etc~~.
15h	**Namespace Insufficient Capacity:** Creating the namespace requires more free space than is currently available. The Command Specific Information field of the Error Information Log specifies the total amount of 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.

Modify a portion of section 5.21.1.1 (Arbitration) as shown below:

5.21 Set Features command

…

5.21.1.1 Arbitration (Feature Identifier 01h)

This Feature controls command arbitration. Refer to section 4.11 for command arbitration details. The attributes are indicated in Command Dword 11.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 136 are returned in Dword 0 of the completion queue entry for that command.

Figure 136: Arbitration & Command Processing – Command Dword 11

Bit	Description
31:24	**High Priority Weight (HPW):** This field defines the number of commands that may be executed from the high priority service class in each arbitration round. This is a 0's based value.
23:16	**Medium Priority Weight (MPW):** This field defines the number of commands that may be executed from the medium priority service class in each arbitration round. This is a 0's based value.
15:08	**Low Priority Weight (LPW):** This field defines the number of commands that may be executed from the low priority service class in each arbitration round. This is a 0's based value.
07:03	Reserved
02:00	**Arbitration Burst (AB):** Indicates the maximum number of commands that the controller may launch at one time from a particular Submission Queue. ~~This value is specified as 2^n.~~ The value is expressed as a power of two (e.g., 000b indicates one, 011b indicates eight). A value of 111b indicates no limit. ~~Thus, the possible settings are 1, 2, 4, 8, 16, 32, 64, or no limit.~~

Modify a portion of section 5.21.1.3 (LBA Range Type) as shown below:

5.21.1.3 LBA Range Type (Feature Identifier 03h), (Optional)

This feature indicates the type and attributes of LBA ranges that are part of the specified namespace. ~~The LBA range information may be used by a driver to determine if it may utilize a particular LBA range; the information is not exposed to higher level software.~~ If multiple Set Features commands for this feature are processed, only information from the most recent successful command is retained (i.e., subsequent commands replace information provided by previous commands).

~~This is optional information that is not required for proper behavior of the system. However, it may be utilized to avoid unintended software issues. For example, if the LBA range indicates that it is a RAID volume then a driver that does not have RAID functionality should not utilize that LBA range (including not overwriting the LBA range). The optional information may be utilized by the driver to determine whether the LBA Range should be exposed to higher level software.~~

The LBA Range Type feature uses Command Dword 11 and specifies the type and attribute information in the data structure indicated in Figure 139. The data structure is 4096 bytes in size and shall be physically contiguous.

If a Get Features command is submitted for this Feature, the attributes specified in Figure 138.a are returned in Dword 0 of the completion queue entry and the LBA Range Type data structure specified in Figure 139 is returned in the data buffer for that command.

Figure 138: LBA Range Type – Command Dword 11

Bit	Description
31:06	Reserved
05:00	**Number of LBA Ranges (NUM):** This field specifies the number of LBA ranges specified in this command. This is a 0's based value. This field is used for the Set Features command only and is ignored for the Get Features command for this Feature.

Figure 138.a: LBA Range Type – Dword 0 of command completion queue entry

Bit	Description
31:06	Reserved
05:00	**Number of LBA Ranges (NUM):** This field indicates the number of valid LBA ranges returned in the data buffer for the command (refer to Figure 139). This is a 0's based value.

Each entry in the LBA Range Type data structure is defined in Figure 139. The LBA Range feature is a set of 64 byte entries; the number of entries is indicated as a command parameter, the maximum number of entries is 64. The LBA ranges shall not overlap and may be listed in any order (e.g., ordering by LBA is not required). If the LBA ranges overlap, the controller should return an error of Overlapping Range.

For a Get Features command, the controller shall clear to zero all ~~All~~ unused entries in the LBA Range Type data structure. ~~shall be cleared to all zeroes for both Get Features and~~ For a Set Features command, the controller shall ignore all unused entries in the LBA Range Type data structure.

The default value for this feature should clear the Number of LBA Ranges field to 00h (i.e., one LBA Range is present) and initialize the LBA Range Type data structure to contain a single entry with the:

- Type field cleared to 00h;
- Attributes field set to 01h;
- Starting LBA field cleared to 0h;
- Number of Logical Blocks field set to indicate the number of LBAs in the namespace; and
- GUID field cleared to 0h, or set to a globally unique identifier.

Figure 139: LBA Range Type – Data Structure Entry

<table>
<tr><th>Byte</th><th>Description</th></tr>
<tr><td>00</td><td>Type (Type): Specifies the Type of the LBA range. The Types are listed below.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>00h</td><td>~~Reserved~~General Purpose</td></tr>
<tr><td>01h</td><td>Filesystem</td></tr>
<tr><td>02h</td><td>RAID</td></tr>
<tr><td>03h</td><td>Cache</td></tr>
<tr><td>04h</td><td>Page / swap file</td></tr>
<tr><td>05h – 7Fh</td><td>Reserved</td></tr>
<tr><td>80h - FFh</td><td>Vendor Specific</td></tr>
</table></td></tr>
<tr><td>01</td><td>Attributes: Specifies attributes of the LBA range. Each bit defines an attribute.
<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>0</td><td>If set to ‘1’, the LBA range may be overwritten. If cleared to ‘0’, the area should not be overwritten.</td></tr>
<tr><td>1</td><td>If set to ‘1’, the LBA range should be hidden from the OS / EFI / BIOS. If cleared to ‘0’, the area should be visible to the OS / EFI / BIOS.</td></tr>
<tr><td>2 – 7</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>15:02</td><td>Reserved</td></tr>
<tr><td>23:16</td><td>Starting LBA (SLBA): This field specifies the 64-bit logical block address of the first logical block that is part of this LBA range.</td></tr>
<tr><td>31:24</td><td>Number of Logical Blocks (NLB): This field specifies the number of logical blocks that are part of this LBA range. This is a 0’s based value (e.g., the value zero specifies one block).</td></tr>
<tr><td>47:32</td><td>Unique Identifier (GUID): This field ~~is~~ contains a global unique identifier, for use by the host, that uniquely specifies the type of this LBA range. Well known Types may be defined and ~~are~~ published on the NVM Express website.</td></tr>
<tr><td>63:48</td><td>Reserved</td></tr>
</table>

~~The host storage driver should expose all LBA ranges that are not set to be hidden from the OS / EFI / BIOS in the Attributes field. All LBA ranges that follow a hidden range shall also be hidden; the host storage driver should not expose subsequent LBA ranges that follow a hidden LBA range.~~

Modify a portion of section 5.21.1.6 (Volatile Write Cache) as shown below:

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

This Feature controls the volatile write cache, if present, on the controller. If a volatile write cache is ~~supported~~ present (refer to the VWC field in Figure 109), then this feature shall be supported. The attributes are indicated in Command Dword 11.

Note: If the controller is able to guarantee that data present in a write cache is written to non-volatile media on loss of power, then that write cache is considered non-volatile and this ~~setting~~ feature does not apply to that write cache. ~~In that case, this setting has no effect.~~

If a Get Features command is submitted for this Feature, the attributes specified in Figure 142 are returned in Dword 0 of the completion queue entry for that command.

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

Figure 1: Volatile Write Cache – Command Dword 11

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

Modify a portion of section 5.21.1.7 (Number of Queues) as shown below:

5.21.1.7 Number of Queues (Feature Identifier 07h)

This Feature indicates the number of queues that the host requests for this controller. This feature shall only be issued during initialization prior to creation of any I/O Submission and/or Completion Queues. If a Set Features command is issued for this feature after creation of any I/O Submission and/or I/O Completion Queues, then the Set Features command shall fail with status code of Command Sequence Error. The value allocated shall not change between resets. For a Set Features command, t~~T~~he attributes are indicated in Command Dword 11 (refer to Figure 143). For a Get Features command, Dword 11 is ignored.

If a Set Features or Get Features command is submitted for this Feature, the attributes specified in Figure 144 are returned in Dword 0 of the completion queue entry for that command.

Figure 143: Number of Queues – Command Dword 11

Bit	Description
31:16	**Number of I/O Completion Queues Requested (NCQR):** Indicates the number of I/O Completion Queues requested by software. This number does not include the Admin Completion Queue. A minimum of one queue shall be requested, reflecting that the minimum support is for one I/O Completion Queue. This is a 0's based value. The maximum value that may be specified is 65,534 (~~indicating~~ i.e., 65,535 I/O Completion Queues). If the value specified is 65,535, the controller should return an error of Invalid Field in Command.
15:00	**Number of I/O Submission Queues Requested (NSQR):** Indicates the number of I/O Submission Queues requested by software. This number does not include the Admin Submission Queue. A minimum of one queue shall be requested, reflecting that the minimum support is for one I/O Submission Queue. This is a 0's based value. The maximum value that may be specified is 65,534 (~~indicating~~ i.e., 65,535 I/O Submission Queues). If the value specified is 65,535, the controller should return an error of Invalid Field in Command.

Note: The value allocated may be smaller or larger than the number of queues requested, often in virtualized implementations. The controller may not have as many queues to allocate as are requested. Alternatively, the controller may have an allocation unit of queues (e.g. power of two) and may supply more queues to host software to satisfy its allocation unit.

Figure 144: Number of Queues – Dword 0 of command completion queue entry

Bit	Description
31:16	**Number of I/O Completion Queues Allocated (NCQA):** Indicates the number of I/O Completion Queues allocated by the controller. A minimum of one queue shall be allocated, reflecting that the minimum support is for one I/O Completion Queue. The value may not match the number requested by host software. This is a 0's based value.
15:00	**Number of I/O Submission Queues Allocated (NSQA):** Indicates the number of I/O Submission Queues allocated by the controller. A minimum of one queue shall be allocated, reflecting that the minimum support is for one I/O Submission Queue. The value may not match the number requested by host software. This is a 0's based value.

Modify a portion of section 5.21.1.13 (Host Memory Buffer) as shown below (changes to Figure 158 are based on changes published in ECN-002):

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

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

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

a) the Host Memory Descriptor List (refer to Figure 156); and
b) the associated host memory region (i.e., the memory regions described by the Host Memory Descriptor List, ~~buffer size, or descriptor list~~

until the host memory buffer has been disabled.

If the host memory buffer is enabled, then a Set Features command to enable the host memory buffer (i.e., the EHM bit (refer to Figure 151) set to '1') shall fail with a status code of Command Sequence Error.

If the host memory buffer is not enabled, then a Set Features command to disable the host memory buffer (i.e., the EHM bit (refer to Figure 151) set to '0') shall succeed without taking any action.

After a successful completion of a Set Features command that disables the host memory buffer, the controller shall not access any data in the host memory buffer until the host memory buffer has been enabled. The controller should retrieve any necessary data from the host memory buffer in use before posting the completion queue entry for the Set Features command that disables the host memory buffer. Posting of the completion queue entry for the Set Features command that disables the host memory buffer acknowledges that it is safe for the host software to modify the host memory buffer contents. Refer to section 8.9.

Figure 151: Host Memory Buffer – Command Dword 11

Bit	Description
31:02	Reserved
01	**Memory Return (MR):** If set to '1', then the host is returning previously allocated memory the controller used prior to a reset or entering the Runtime D3 state. A returned host memory buffer shall have the exact same size, descriptor list address, descriptor list contents, and host memory buffer contents as last seen by the controller before the host memory buffer was disabled (i..e., a Set Features command with the EHM bit ~~was~~ cleared to '0' was processed). If cleared to '0', then the host is allocating host memory resources with undefined content.
00	**Enable Host Memory (EHM):** If set to '1', then the host memory buffer shall be enabled and the controller may use the host memory buffer. ~~While~~ If cleared to '0', then the host memory buffer shall be disabled, and the controller shall not use the host memory buffer. If a Set Features command is processed with this bit cleared to '0', then the controller shall ignore Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15.

…

If a Get Features command is issued for this Feature, the attributes specified in Figure ~~151~~157.a are returned in Dword 0 of the completion queue entry and the Host Memory Buffer Attributes data structure, whose structure is defined in Figure 158, is returned in the data buffer for that command.

Figure 157.a: Host Memory Buffer – Dword 0 of command completion queue entry

Bit	Description
31:01	Reserved
00	**Enable Host Memory (EHM):** If set to '1', then the host memory buffer is enabled and the controller may use the host memory buffer. If cleared to '0', then the host memory buffer is disabled, and the controller is not using the host memory buffer.

Figure 158: Host Memory Buffer – Attributes Data Structure

Byte	Description
3:0	**Host Memory Buffer Size (HSIZE):** This field indicates the size of the host memory buffer allocated in memory page size units.
7:4	**Host Memory Descriptor List Address Lower (HMDLAL):** This field indicates the lower 32 bits of the physical location of the Host Memory Descriptor List (refer to Figure 156) for the ~~Host~~ host ~~M~~memory ~~B~~buffer. This address shall be 16 byte aligned. The lower 4 bits shall be cleared to zero.
11:8	**Host Memory Descriptor List Address Upper (HMDLAU):** This field indicates the upper 32 bits of the physical location of the Host Memory Descriptor List (refer to Figure 156) for the ~~Host~~ host ~~M~~memory ~~B~~buffer.
15:12	**Host Memory Descriptor List Entry Count (HMDLEC):** This field indicates the number of valid Host Memory Descriptor Entries (refer to Figure 157) in the Host Memory Descriptor List (refer to Figure 156).
4095:16	**Reserved**

Modify a portion of section 5.21.1.17 (Non-Operational Power State Config) as shown below:

5.21.1.17 Non-Operational Power State Config (Feature Identifier 11h), (Optional)

This Feature configures non-operational power state settings for the controller. The settings are specified in Command Dword 11.

If a Get Features command is submitted for this Feature, the values in Figure 163 are returned in Dword 0 of the completion queue entry for that command.

Figure 163: Non-Operational Power State Config – Command Dword 11

Bit	Description
31:01	Reserved
00	**Non-Operational Power State Permissive Mode Enable (NOPPME):** If NOPPME is set to '1' then the controller may temporarily exceed the power limits of any non-operational power state, up to the limits of the last operational power state, to run controller initiated background operations in that state (i.e., Non-Operational Power State Permissive Mode is enabled). If NOPPME is cleared to '0', then the controller shall not exceed the limits of any non-operational state while running controller initiated background operations in that state (i.e., Non-Operational Power State Permissive Mode is disabled). Performance may be degraded if Non-Operational Power Permissive Mode is disabled. If the host attempts to set this field to '1' and the controller does not support Non-Operational Power State Permissive Mode as indicated in the Controller Attributes field of Identify Controller, then the command fails with a status of Invalid Field in Command.

Modify a portion of section 5.23 (Format NVM command) and 5.23.1 (Command Completion) as shown below (these changes are based on changes published in ECN-003):

5.23 Format NVM command – NVM Command Set Specific

…

Figure 2: Format NVM – Format Scope

FNA[1] Bit 0	NSID	Format Operation
0b	FFFFFFFFh	All namespaces attached to the controller
0b	~~All other~~ Any valid value~~s~~ (refer to section 6.1.2)	Particular namespace specified
1b	~~All~~ Any valid value~~s~~ (refer to section 6.1.2) or FFFFFFFFh	All namespaces in the NVM subsystem

NOTES:

1. FNA is the Format NVM Attributes field in the Identify Controller data structure.

Figure 3: Format NVM – Secure Erase Scope

FNA[1] Bit 1	NSID	Secure Erase
0b	FFFFFFFFh	All namespaces attached to the controller
0b	~~All other~~ Any valid value~~s~~ (refer to section 6.1.2)	Particular namespace specified
1b	~~All~~ Any valid value~~s~~ (refer to section 6.1.2) or FFFFFFFFh	All namespaces in the NVM subsystem
NOTES: 1. FNA is the Format NVM Attributes field in the Identify Controller data structure.		

The Format NVM command shall fail if the controller is in an invalid security state (refer to the appropriate security specification, e.g., TCG ~~SIIS~~ Storage Interface Interactions Specification). The Format NVM command may fail if there are outstanding I/O commands to the namespace specified to be formatted. I/O commands for a namespace that has a Format NVM command in progress may fail.

…

5.23.1 Command Completion

A completion queue entry is posted to the Admin Completion Queue when the NVM media format is complete. Format NVM command specific status values are defined in Figure 177.

Figure 177: Format NVM – Command Specific Status Values

Value	Description
Ah	**Invalid Format:** The format specified is invalid. This may be due to various conditions, including: 1) specifying an invalid LBA Format number, or 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 ~~SIIS~~ Storage Interface Interactions Specification), etc.

Modify a portion of section 6.7.1.1 (Deallocate) as shown below:

6.7 Dataset Management command

…

6.7.1.1 Deallocate

…

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

- have the Guard field value set to FFFFh or set to the CRC for the value read from the deallocated logical block and its metadata (excluding protection information) (e.g., set to 0000h if the value read is all bytes set to 00h); and
- have the Application Tag field value set to FFFFh and the Reference Tag field value set to FFFFFFFFh (indicating the protection information shall not be checked).

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

Modify a portion of section 6.8 (Flush command) as shown below:

6.8 Flush command

The Flush command is used to request that the contents of volatile write cache be made non-volatile.

If a volatile write cache is enabled (refer to section 5.21.1.6) then ~~T~~the Flush command shall commit data and metadata associated with the specified namespace(s) to non-volatile media. The flush applies to all commands completed prior to the submission of the Flush command. The controller may also flush additional data and/or metadata from any namespace.

If a volatile write cache is not present or not enabled, then Flush commands shall complete successfully and have no effect.

All command specific fields are reserved.

6.8.1 Command Completion

~~If the command is completed, then the controller shall post a completion queue entry to the associated I/O Completion Queue indicating the status for the command.~~ Upon completion of the Flush command, the controller posts a completion queue entry to the associated I/O Completion Queue.

Modify a portion of section 7.10 (Identifier Format and layout) to add new section 7.10.6 as shown below:

7.10 Identifier Format and Layout (Informative)

This section provides guidance for proper implementation of various identifiers defined in the Identify Controller, ~~and~~ Identify Namespace, and Namespace Identification Descriptor data structures.

…

7.10.6 Universally Unique Identifier (UUID)

The Universally Unique Identifier is defined in RFC4122 and contained in the Namespace Identification Descriptor (refer to Figure 116). Byte ordering requirements for a UUID are described in RFC4122.

Modify a portion of section 7.11 (Unique Identifier) as shown below (changes are based on changes published in ECN-002):

7.11 Unique Identifier

…

The Identify Namespace data structure (refer to Figure 114) contains the IEEE Extended Unique Identifier (EUI64) and the Namespace Globally Unique Identifier (NGUID) fields. The Namespace Identification Descriptor data structure (refer to Figure 116) contains the Namespace UUID. EUI64 is an 8-byte EUI-64 identifier (refer to section 7.10.4), ~~and~~ NGUID is a 16-byte identifier based on EUI-64 (refer to section 7.10.5), and Namespace UUID is a 16-byte identifier described in RFC4122 (refer to 7.10.6).

When creating a namespace, the controller ~~specifies~~ shall indicate a globally unique value in one or more of the following:

a) the EUI64 field;
b) the NGUID field; or
c) a Namespace Identification Descriptor with the Namespace Identifier Type field set to 3h ~~or NGUID field (the controller may optionally specify a globally unique value in both fields)~~.

~~In cases where the 64-bit EUI64 field is unable to ensure a globally unique namespace identifier, the EUI64 field shall be cleared to 0h. When not implemented, these fields contain a value of 0h.~~ If the EUI64 field is cleared to 0h and the NGUID field is cleared to 0h, then the namespace shall support a valid Namespace UUID in the Namespace Identification Descriptor data structure.

If bit 3 in NSFEAT 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 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.

Add new section 7.14 (Privileged Actions) as shown below:

7.14 Privileged Actions

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

Admin commands that are privileged include Namespace Management, Namespace Attachment, Virtualization Management, Format NVM, and Sanitize. A privileged register action is NVM subsystem reset. Vendor specific commands and registers may also be privileged.

Modify a portion of section 8.4.1 (Non-Operational Power States) as shown below:

8.4 Power Management

…

8.4.1 Non-Operational Power States

…

While in a non-operational state, a controller may exceed the power advertised by the state for the following purposes:

- servicing a memory-mapped I/O (MMIO) or configuration register access (e.g., Controller Configuration (refer to section 3.1.5));
- processing a command submitted to the Admin Submission Queue and processing background operations, if any, initiated by that command (~~such as~~ e.g., Device Self-test command (refer to section 5.8), Sanitize command (refer to section 5.24)); or
- if Non-Operational Power State Permissive Mode is supported and enabled, executing controller initiated background operations (refer to section 5.21.1.17).

For all of the cases in the preceding paragraph, the controller shall:

- logically remain in the current non-operational power state unless an IO command is received or if an explicit transition is requested by a Set Features command with the Power Management identifier; and
- not exceed the maximum power advertised for the most recent operational power state.

Modify a portion of section 8.4.4 (Runtime D3 Transitions) as shown below:

8.4.4 Runtime D3 Transitions

…

The RTD3 Resume Latency is ~~measured~~ the expected elapsed time from the time power is applied until the controller is able to~~: complete an I/O command~~

a) process and complete I/O commands; and
b) access the NVM associated with attached namespace(s), if any, as part of I/O command processing.

The latency reported is based on a normal shutdown with optimal controller settings preceding the RTD3 resume. The latency reported assumes that host software enables and initializes the controller and ~~then~~ sends a 4KB read operation.

The RTD3 Entry Latency is ~~measured~~ the expected elapsed time from the time CC.SHN is set to 01b by host software until ~~CC~~CSTS.SHST is set to 10b by the controller. When ~~CC~~CSTS.SHST is set to 10b, it is safe for host software to remove power from the controller.

Modify a portion of section 8.5 (Virtualization Enhancements) as shown below:

8.5 Virtualization Enhancements (Optional)

…

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

Modify a portion of section 8.8.2 (Registering) as shown below:

8.8 Reservations (Optional)

…

8.8.2 Registering

Prior to establishing a reservation on a namespace, a host shall become a registrant of that namespace by registering a reservation key. This reservation key may be used by the host as a means of identifying the registrant (host), authenticating the registrant, and preempting a failed or uncooperative registrant. The value of the reservation key used by a host and the method used to select its value is outside the scope of this specification.

Registering a reservation key with a namespace creates an association between a host and a namespace. A host that is a registrant of a namespace may use any controller with which ~~it~~ that host is associated (i.e., that has the same Host Identifier, refer to section 5.21.1.19) to access that namespace as a registrant. Thus, a host need only register on a single controller ~~in order~~ to become a registrant of the namespace on all controllers in the NVM subsystem that have access to the namespace and are associated with the host.

A host registers a reservation key by executing a Reservation Register command on the namespace with the Reservation Register Action (RREGA) field set to 000b (i.e., Register Reservation Key) and supplying a reservation key in the New Reservation Key (NRKEY) field.

A host that is a registrant of a namespace may register the same reservation key value multiple times with the namespace on the same or different controllers. ~~It is an error for a host that is already a registrant of a namespace to register with the same namespace using a different registration key value (i.e., the command is aborted with status Reservation Conflict).~~ For a Reservation Register command with the RREGA field set to 000b:

a) the IEKEY field shall be ignored; and
b) if a host that is already a registrant of a namespace attempts to register with that namespace using a different registration key value, then the command shall be aborted with status Reservation Conflict.

There are no restrictions on the reservation key value used by hosts with different Host Identifiers. For example, multiple hosts may all register with the same reservation key value.

A host that is a registrant of a namespace may replace ~~its~~ the existing reservation key value for that namespace by executing a Reservation Register command on the namespace with the:

a) RREGA field set to 010b (i.e., Replace Reservation Key)~~,~~;
b) ~~supplying the~~ current reservation key in the Current Reservation Key (CRKEY) field~~,~~; and
c) ~~the~~ new reservation key in the NRKEY field.

The current reservation key value shall be replaced by the new reservation key value in all controllers to which the namespace is attached that have the same Host identifier as the Host identifier of the controller processing the command. If the contents of the CRKEY field do not match the key currently associated with the host, then the command ~~is~~ shall be aborted with a status of Reservation Conflict. A host may replace its reservation key without regard to its registration status or current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the Reservation Register command. Replacing a reservation key has no effect on any reservation that may be held on the namespace.

Modify a portion of section 8.10 (Replay Protected Memory Block) as shown below:

8.10 Replay Protected Memory Block (Optional)

…

Figure 272: RPMB Contents

Content	Type	Size	Description
Authentication Key	Write once, not erasable or readable	Size is dependent on authentication method reported in Identify Controller data structure (e.g. SHA-256 is 32 bytes (refer to RFC 6234))	Authentication key which is used to authenticate accesses when MAC is calculated.
…			

…

Figure 273: RPMB Data Frame

Bytes	Component Name	Description
222-*N*:00	Stuff Bytes	Padding for the frame. Values in this field are not part of the MAC calculation. The size is 223 bytes minus the size of the Authentication Key (*N*).
222:222-(*N*-1)	Authentication Key or Message Authentication Code (MAC)	Size is dependent on authentication method reported in the Identify Controller data structure (e.g., SHA-256 key is 32 bytes (refer to RFC 6234)).
…		

Modify a portion of section 8.10.1 (Authentication Method) as shown below:

8.10.1 Authentication Method

A controller supports one Authentication Method as indicated in the Identify Controller data structure.
If the Authentication Method supported is HMAC SHA-256 (refer to RFC 6234), then the message authentication code (MAC) is calculated using HMAC SHA-256 as defined in ~~[HMAC-SHA]~~ RFC 6234. The key used to generate a MAC using HMAC SHA-256 is the 256-bit Authentication Key stored in the controller for the selected RPMB target. The HMAC SHA-256 calculation takes as input a key and a message. Input to the MAC calculation is the concatenation of the fields in the RPMB Data Frame (request or response) excluding stuff bytes and the MAC itself – i.e., bytes [223:255] and Data of the frame in that order.