LEGAL NOTICE:

© Copyright 2008 to 2022 NVM Express, Inc. ALL RIGHTS RESERVED.
This erratum is proprietary to the NVM Express, Inc. (also referred to as “Company”) and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this erratum 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: **“© 2008 to 2022 NVM Express, Inc. ALL RIGHTS RESERVED.”** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

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

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

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

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

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

NVM Express® Technical Errata

Errata ID	**105**
Revision Date	**10/03/2022**
Affected Spec Ver.	**NVM Express® Base Specification Revision 2.0b** **NVM Express® NVM Command Set Specification Revision 1.0b** **NVM Express® Zoned Namespace Command Set Specification Revision 1.1b** **NVM Express® RDMA Transport Specification Revision 1.0a** **NVM Express® TCP Transport Specification Revision 1.0b** **NVM Express® Management Interface Revision 1.2b**
Corrected Spec Ver.	

Errata Author(s)

Name	**Company**
Fred Knight	NetApp
Judy Brock, Mike Allison	Samsung
Paul Suhler, John Geldman	Kioxia
Kalyan Sanagavarapu, Myron Loewen	Intel
Myron Loewen	Solidigm Technology
Curtis Ballard	HPE
Austin Bolen, David Black	Dell EMC

Errata Overview

This ECN updates and clarifies various text within the NVM Express Base Specification, the NVM Express NVM Command Set Specification, the NVM Express Zoned Namespace Command Set Specification, the NVM Express RDMA Transport Specification, the NVM Express TCP Transport Specification and the. NVM Express Management Interface Specification.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Revision History

Revision Date	Change Description
5/24/2021	Initial creation
7/22/2021	Add more email inputs
1/05/2022	Prep for TWG presentation (remove some duplicate material and duplicate notes)
1/21/2022	Update to ‘a’ / ‘b’ versions of specs as appropriate.
1/24/2022	Update Change list descriptions; fix “compliant” language.
2/24/2022	Completed TWG review, add sentence in Format and Sanitize to replace Write Protect error text.
2/25/2022	Incorporate review comments from David Black (with change tracking enabled)
2/28/2022	Incorporate review comments from Judy Brock (with change tracking enabled)
3/2/2022	Incorporate review comments from Peter Onufryk (with change tracking enabled)
3/3/2022	Incorporate review comments from TWG review (tracked changes accepted)
3/15/2022	Update WRITE ZEROES description to be explicit about read requirements if a deallocate actually happens. Update CSS table cell to add back missing text.
4/14/2022	Update from 30-day review on Phase Tag description to account for differences in Admin Queue initialization (setting CC.EN=1) and I/O Queue initialization (Create I/O Submission Queue command).
4/19/2022	Update error returned (Figure 380) when creating a queue with a QID that already exists. This change requires a restart of 30 day review.
5/31/2022	Update Memory-Based Transport Model section based on editorial comments from 30-day review.
6/9/2022	Accept tracked changes for integration.
8/9/2022	Integrated
8/30/2022	Incorporate review comments from Mike Allison and Fred Knight
9/27/2022	Incorporate review comments from Mike Allison and Fred Knight
10/3/2022	Update “NVM power state” to “NVMe power state” in Figure 175 of NVM Express Management Interface Specification Appendix A

Description of Changes

NVM Express Base specification:

Clarify multiple instances where “command” was added to distinguish cases where something other than a command was being referred to.

Update descriptions and references that discuss the multiple “Identify Namespace data structures” (1.5.29).

Create a specific section to describe the “P” bit in the CQE, along with multiple references to that new section (3.3.3.2.2).

Update name of NVMe Management Interface specification to match the actual spec – NVM Express Management Interface Specification.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Correct multiple upper/lower case usage for Command Set specification names.

Update descriptions of spec revision compliance requirements (change from 1.4 based description to 2.0 based description). There are no changes to the requirements.

Clarify description of the CC.CSS field (change from text-based description to a table-based description).

Clarify that NSID, NVM Set ID, and Endurance Group ID value uniqueness requirements are scoped to the NVM subsystem.

Add Create I/O Submission Queue to the list of commands that are NVM Set aware (figure 73). The command was always NVM Set aware but had inadvertently been left out of this list.

Update the Controller initialization flows for the host to indicate that Identify Controller operations are per command set, rather than per namespace.

Correct multiple descriptions to use the standard wording “controller shall abort the command with a status code of …” rather than “status code of … is returned”. This clarifies that the command is in fact aborted.

Update several “refer to” links that were incorrect; including some pointers to steps in ordered lists.

Update language “compliant to” to be “compliant with” (“comply with” or “compliant with” are correct (as is “conform to”), but “comply / compliant to” is incorrect).

Update the Capacity Adjustment Factor value for the example in section 3.8.2.3 to be more appropriate for that example.

Specifically prohibit data transfers that cause buffer overrun (as some text implied it was required).

Correct errors in Figure 122 (section 4.1.2.1) that were introduced by the “KB” to “KiB” changes in revision 1.4 (correct size of the “A” descriptor; correct the address pointer for segment “A” and “B”).

Update references to the “NVMe over Fabrics” specification to point to the correction section of the NVMe Base specification.

Remove duplicate text from the Asynchronous Event Information description.

Remove multiple instances (across multiple commands) of “Namespace is Write Protected” from the Command Specific status value lists. While this error is valid for those commands, this error, is a Generic Command Status value (SCT=0h), and not a Command Specific status value (SCT=1h).

Update the description for the “Invalid Firmware Image” error to include the case where a firmware slot does not contain a firmware image.

Remove “Command Sequence Error” and “Operation Denied” from the Command Specific status value lists. While this error is valid for those commands, this error, is a Generic Command Status value (SCT=0h), and not a Command Specific status value (SCT=1h).

Remove “Access Denied” from the Command Specific status value lists. While this error is valid for those commands, this error, is a Media and Data Integrity error value (SCT=2h), and not a Command Specific status value (SCT=1h).

Add references in the Get Features command to the controller type specific lists of mandatory, optional, and prohibited Feature Identifiers.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Clarify that the RAE bit (Figure 197) actions occur immediately upon successful completion of the command and not at some random time “after” the completion of the command.

Correct the bit value description for the Get Log Page data structure (Figure 204). Bits 16 to 31 had been listed in the description twice.

Simplify and clarify the description of the “Establish Context and Read 512 Bytes of Header Supported” bit (Figure 205).

Clarify the description of the Parameter Error Location field in Log Identifier 01h, and include the relationship between the bit location and the byte location.

Clarify the timing associated with the logging of Critical Warning information in the SMART / Health Information Log (Figure 207).

Clarify that the “I/O Command Set Independent Identify Namespace data structure is part of the “Identify Namespace data structures” included in Changed Namespace List notifications (section 5.16.1.5).

Clarify the timing associated with the logging of Critical Warning information in the Endurance Group Information Log (Figure 217).

Explicitly state that if no namespaces are attached to a controller, the ANA log page does not contain any descriptors. This was previously already implied by other requirements but is now explicitly stated.

Correct the reference in Persistent Event Log (5.16.1.14) for sanitize operation interactions.

Clarify port identifier descriptions for the persistent error log (section 5.16.1.14).

Clarify the timing related to adding Endurance Group Critical Warning Condition events to the Hardware Error Event type entry.

Clarify the description of the Status field for the Format NVM Command Completion event data (Figure 238). This was updated to explicitly match all other instances of use of the Status field in other data structures. The revision field for the data structure was updated to 02h to indicate that the field does use the defined format. Due to the previous lack of clarity, this may be considered by some to be an incompatible technical change.

Clarify Format NVM command and Sanitize command text for cases with Write Protected namespaces.

Clarify and correct the formula used to calculate the offset in the Set Feature Event Data Format data structure for the Set Feature Event type log (Figure 241).

Clarify ordering requirements for sub-structures embedded in the Configuration Descriptors log page (section 5.16.1.17).

Clarify that Identify Controller fields that are Fabric Specific do not apply to memory-based transport controllers.

Correct Namespace Management command error “Invalid Format” for the TCG specific cases (See the TCG specification for error cases).

Clarify several Fabric commands response information for their Status field (to match other generic requirements for this field for other commands). Bits 1 to 15 contain the Status field, and bit 0 is reserved (for Fabric); bit 0 is the “P” bit for memory based transports but is reserved for Fabric transports.

Clarify that a Connect command that attempts to create a duplicate queue with the same QID is aborted with the status code of Command Sequence Error.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Add several reference links in the Reservation command sections to get directly to the specific actions for that particular Reservation command.

Clarify that the “512 byte units” used for Telemetry means 512 byte multiple for the starting offset and length. The error cases for non-512 multiples are already described in other parts of the specification.

Clarify that the term “controller-initiated log” is referring to the “Telemetry Controller-Initiated log page” and that the term “host-initiated log” is referring to the “Telemetry Host-Initiated log page”.

Correct the reference in the Asynchronous Event Request Host Annex (section B.4) to refer to the correct step in the preceding list.

Incompatible Changes:

The Status field (bytes 11:08) in the Format NVM Completion Event (see section 5.16.1.14.1.8) did not describe where in that 32-bit field to put the 16 valid bits. That was corrected by matching the description of this field to all other instances of reporting of the Status field information. This may be considered an incompatible change.

NVM Express NVM Command Set:

Add NVM Express Management Interface Specification to the references section (so the “refer to” statements can be resolved).

Update definitive statements (when) to conditional statements (if); because it really is conditional.

Add cross references to as appropriate.

Update language “compliant to” to be “compliant with” (“comply with” or “compliant with” are correct (as is “conform to”), but “comply / compliant to” is incorrect).

Remove multiple instances (across multiple commands) of “Namespace is Write Protected” from the Command Specific status value lists. While this error is valid for those commands, this error, is a Generic Command Status value (SCT=0h), and not a Command Specific status value (SCT=1h).

Simplify and clarify description of the DEAC bit in the Write Zeroes command.

Clarify that the PIP field indicates the placement of the protection information and add reference to this bit in the model that describes protection information placement.

Add host recommendation for the PRACT bit to the Write Zeroes command description for namespaces formatted with protection information (section 3.2.8).

Update Figure 105 for ENDGID for the ‘0’ value (this was documented in the Endurance Group section but was not listed in this table). This ensures a consistent description of the value ‘0’.

Correct the reference for the 64b CRC polynomial calculation.

NVM Express Zoned Namespace Command Set:

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Update description of Write Pointer. Eliminate the possibility of a time based (next) interpretation by using a location based (lowest number LBA).

Clarify (reword) the different results that are possible (during resource shortages) dependent on the number of zones that are open in the Implicitly Opened state.

Use consistent terminology for “active zones” and “open zones”.

Clarify setting of “Reset Zone Recommended” attribute state for Read only and Offline zones.

Clarify setting of “Finish Zone Recommended” attribute state for Read only and Offline zones.

Clarify setting of “Zone Finished by Controller” attribute state for Read only and Offline zones.

NVM Express® RDMA Transport:

Add the correct reference to the InfiniBand Specification.

Update definitions with references into the InfiniBand Specification so the corresponding terms can be found in that document.

Clarify the usage of the CRQSIZE field in the RDMA_CM_ACCEPT message.

NVM Express TCP Transport:

Correct the heading for Figure 3 to indicate that the example shows 16B PDU alignment.

NVM Express® Management Interface:

Clarify usage of CIAP bit and MEB bit in NVMe-MI Message fields.

Clarify wording for the NVM Subsystem Power (NVMSP) field.

Editor’s Note:
BLACK text indicates unchanged text; **BLUE** text indicates newly inserted text, **RED** text indicates deleted text; **PURPLE** text indicates moved text, **GREEN** text indicates editor notes.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Description of NVM Express® Base Specification changes

<Note to Editor: Global change: "Invalid Field In Command" should be changed to "Invalid Field in Command". Note: Change upper case "In" to lower case "in" to match the definition of the term.>

Modify a portion of section 1 as shown below:

1.5.33 I/O Submission Queue

An I/O Submission Queue is a Submission Queue that is used to submit I/O commands for execution by the controller (e.g., Read command and Write command for the NVM ~~command set~~ Command Set).

…

Modify a portion of section 2 as shown below:

2 Theory of Operation

…

A namespace is a formatted quantity of non-volatile memory that may be accessed by a host. Associated with each namespace is an I/O Command Set that operates on that namespace. An NVM Express controller may support multiple namespaces that are referenced using a namespace ID. Namespaces may be created and deleted using the Namespace Management command and Capacity Management command~~s~~. The Identify Namespace data structures (refer to section 1.5.29)~~, the I/O Command Set specific Identify Namespace data structure, and the I/O Command Set Independent Identify Namespace data structure~~ indicate capabilities and settings that are specific to a particular namespace.

…

Modify a portion of section 2.1 as shown below:

2.1 Memory-Based Transport Model

…

A Completion Queue (CQ) is a circular buffer with a fixed slot size used to post status for completed commands. A completed command is uniquely identified by a combination of the associated SQ identifier and command identifier that is assigned by host software. In the memory-based transport model multiple Submission Queues may be associated with a single Completion Queue. ~~This feature~~ A configuration with a single Completion Queue may be used where a single worker thread processes all command completions via one Completion Queue even when those commands originated from multiple Submission Queues. The CQ Head pointer is updated by host software after processing completion queue entries indicating the last free CQ slot. A Phase Tag (P) bit is defined in the completion queue entry to indicate whether an entry has been newly posted without the host consulting a register (refer to section 3.3.3.2.2). ~~This~~ The Phase Tag bit enables the host ~~software~~ to determine whether entries are new or not. ~~the new entry was posted as part of the previous or current round of completion notifications. Specifically, each round through the completion queue entries, the controller inverts the Phase Tag bit~~.

…

Modify a portion of section 2.3 as shown below:

2.3 NVM Storage Model

…

2.3.2 I/O Command Sets

I/O commands perform operations on namespaces, and each namespace is associated with exactly one I/O command set. For example, commands in the NVM ~~command set~~ Command Set access data represented in a namespace as logical blocks, and commands in the Key Value Command ~~set~~ Set access data represented in a namespace as key-value pairs.

The association of a namespace to an I/O command set is specified when the namespace is created and is fixed for the lifetime of the namespace.

…

Modify a portion of section 3.1 as shown below:

<Editors note: Global change from “NVMe Management Interface specification” to “NVM Express Management Interface Specification”. Do NOT change the instance of “NVMe Management Interface” (without the word “specification”).>

3.1.2.2 Administrative Controller

…

Examples of management capabilities that may be supported by an Administrative controller include the following.

- Ability to efficiently poll NVM subsystem health status via NVMe-MI using the NVMe-MI Send command and the NVMe-MI Receive command~~s~~ (refer to the NVM Subsystem Health Status Poll section in the NVM~~e~~ Express Management Interface ~~s~~Specification);
- Ability to manage an NVMe enclosure via NVMe-MI using the NVMe-MI Send command and the NVMe-MI Receive command~~s~~;
- Ability to manage NVM subsystem namespaces using the Namespace Attachment command and the Namespace Management command~~s~~;
- Ability to perform virtualization management using the Virtualization Management command; ~~and~~
- Ability to reset an entire NVM subsystem using the NVM Subsystem Reset (NSSR) register, if supported~~.~~; and
- Ability to shutdown an entire NVM subsystem using the NVM Subsystem Shutdown (NSSD) property, if supported.

…

3.1.3.1 Offset 0h: CAP – Controller Capabilities

…

Figure 36: Offset 0h: CAP – Controller Capabilities

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>63: 61</td><td>RO</td><td>0h</td><td>Reserved</td></tr>
<tr><td>60:59</td><td>RO</td><td>Impl Spec</td><td><b>Controller Ready Modes Supported (CRMS):</b> This field indicates the ready capabilities of the controller. Refer to sections 3.5.3 and 3.5.4 for more detail.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>0</td><td><b>Controller Ready With Media Support (CRWMS):</b> If this bit is set to ‘1’, then the controller supports the Controller Ready With Media mode.
If this bit is cleared to ‘0’, then the controller does not support Controller Ready With Media mode.
This bit shall be set to ‘1’ on controllers compliant with ~~versions later than~~ NVM Express Base Specification, R~~r~~evision ~~1.4~~ 2.0 and later.</td></tr>
<tr><td>1</td><td><b>Controller Ready Independent of Media Support (CRIMS):</b> If this bit is set to ‘1’, then the controller supports the Controller Ready Independent of Media mode.
If this bit is cleared to ‘0’, then the controller does not support Controller Ready Independent of Media mode.</td></tr>
</table>
</td></tr>
<tr><td>…</td><td></td><td></td><td></td></tr>
<tr><td>31:24</td><td>RO</td><td>Impl Spec</td><td><b>Timeout (TO):</b> This is the worst-case time that host software should wait for CSTS.RDY to transition from:
<b>…</b>
Controllers that support the CRTO register (refer to Figure 62) are able to indicate larger timeouts for enabling the controller. Host software should use the value in CRTO.CRWMT or CRTO.CRIMT depending on the controller ready mode indicated by CC.CRIME to determine the worst-case timeout for CSTS.RDY to transition from ‘0’ to ‘1’ after CC.EN transitions from ‘0’ to ‘1’. Host software that is based on revisions earlier than NVM Express Base Specification, R~~r~~evision ~~1.4~~ 2.0 ~~or earlier~~ is not required to wait for more than 127.5 seconds for CSTS.RDY to transition.
Refer to sections 3.5.3 and 3.5.4 for more information.</td></tr>
<tr><td>…</td><td></td><td></td><td></td></tr>
</table>

…

3.1.3.5 Offset 14h: CC – Controller Configuration

…

Figure 46: Offset 14h: CC – Controller Configuration

Bits	Type	Reset	Description
…			

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 46: Offset 14h: CC – Controller Configuration

<table>
<tr><th>Bits</th><th>Type</th><th>Reset</th><th>Description</th></tr>
<tr><td>06:04</td><td>RW</td><td>000b</td><td>

I/O Command Set Selected (CSS): This field specifies the I/O Command Set or Sets that are selected. This field shall only be changed when the controller is disabled (i.e., CC.EN is cleared to ‘0’). The I/O Command Set or Sets that are selected shall be used for all I/O Submission Queues.

~~If CAP.CSS bit 0 is set to ‘1’, then a value of 000b selects the NVM Command Set. If CAP.CSS bit 0 is cleared to ‘0’, then the value 000b is reserved.~~

~~If CAP.CSS bit 6 is set to ‘1’, then a value of 110b selects all I/O Command Sets supported by the controller. The I/O Command Sets that are supported in this case are reported in the Identify I/O Command Set data structure (refer to section 5.17.2.21). If CAP.CSS bit 6 is cleared to ‘0’, then the value 110b is reserved.~~

~~If CAP.CSS bit 7 is set to ‘1’, then the value 111b indicates that only the Admin Command Set is supported and that no I/O Command Set or I/O Command Set specific Admin commands are supported. When only the Admin Command Set is supported, any command submitted on an I/O Submission Queue and any I/O Command Set specific Admin command submitted on the Admin Submission Queue is completed with status Invalid Command Opcode. If CAP.CSS bit 7 is cleared to ‘0’, then the value of 111b is reserved.~~

<table>
<tr><th>Value</th><th colspan="2">Definition</th></tr>
<tr><td rowspan="3">000b</td><th>CAP.CSS bit 0</th><th>Description</th></tr>
<tr><td>1</td><td>NVM Command Set</td></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td>001b to 101b</td><td colspan="2">Reserved</td></tr>
<tr><td rowspan="3">110b</td><th>CAP.CSS Bit 6</th><th>Description</th></tr>
<tr><td>1</td><td>All supported I/O Command Sets

The I/O Command Sets that are supported are reported in the Identify I/O Command Set data structure (refer to section 5.17.2.21).</td></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td rowspan="3">111b</td><th>CAP.CSS Bit 7</th><th>Description</th></tr>
<tr><td>1</td><td>Admin Command Set only

I/O Command Set and I/O Command Set specific Admin commands are not supported. Any I/O Command Set specific Admin command submitted on the Admin Submission Queue is aborted with a status code of Invalid Command Opcode.</td></tr>
<tr><td>0</td><td>Reserved</td></tr>
</table>

For Discovery controllers, this property shall be cleared to 000b.

</td></tr>
<tr><td>…</td><td></td><td></td><td></td></tr>
</table>

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Modify a portion of section 3.2 as shown below:

3.2 NVM Subsystem Entities

…

3.2.1.6 NSID and Namespace Usage

If Namespace Management (refer to section 8.11), ANA Reporting (refer to section 8.1), or NVM Sets (refer to section 3.2) capabilities are supported, then NSIDs shall be unique within the NVM subsystem (e.g., NSID of 3 shall refer to the same physical namespace regardless of the accessing controller). If the Namespace Management, ANA Reporting, and NVM Sets capabilities are not supported, then NSIDs:

a) for shared namespaces shall be unique within the NVM subsystem; and
b) for private namespaces are not required to be unique within the NVM subsystem.

…

3.2.2 NVM Sets

…

Figure 73: NVM Set Aware Admin Commands

Admin Command	Details
…	
Get Features and Set Features	• The Read Recovery Level Feature specifies the associated NVM Set Identifier. • The Predictable Latency Mode Config Feature specifies the associated NVM Set Identifier. • The Predictable Latency Mode Window Feature specifies the associated NVM Set Identifier.
Create I/O Submission Queue	• The Create I/O Submission Queue command includes the associated NVM Set Identifier.

…

An NVM Set Identifier is a 16-bit value that specifies the NVM Set with which an action is associated. An NVM Set Identifier is unique with the NVM subsystem. An NVM Set Identifier may be specified in NVM Set aware Admin commands (refer to Figure 73). An NVM Set Identifier value of 0h is reserved and is not a valid NVM Set Identifier. Unless otherwise specified, if the host specifies an NVM Set Identifier cleared to 0h for a command that requires an NVM Set Identifier, then that command shall abort with a status code of Invalid Field in Command.

…

3.2.3 Endurance Groups

…

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

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Modify a portion of section 3.3 as shown below:

3.3 NVM Queue Models

…

3.3.1.2 Queue Usage

…

Host software checks completion queue entry Phase Tag (P) bits in memory to determine whether new completion queue entries have been posted (refer to section 3.3.3.2.2). The Completion Queue Tail pointer is only used internally by the controller and is not visible to the host. The controller uses the SQ Head Pointer (SQHD) field in completion queue entries to communicate new values of the Submission Queue Head Pointer to the host. A new SQHD value indicates that submission queue entries have been consumed, but does not indicate either execution or completion of any command. Refer to section 3.3.3.2.

…

A completion queue entry is posted to the Completion Queue when the controller write of that completion queue entry to the next free Completion Queue slot inverts the Phase Tag (P) bit from its previous value in memory (refer to section 3.3.3.2.2). The controller may generate an interrupt to the host to indicate that one or more completion queue entries have been posted.

…

3.3.2.3 Queue Initialization and Queue State

When a Connect command successfully completes, the corresponding Admin Submission and Completion Queue or I/O Submission and Completion Queues are created. If the host sends a Connect command specifying the Queue ID of a queue which already exists, then the controller shall abort the command with a status code of Command Sequence Error ~~is returned~~.

The Authentication Requirements (AUTHREQ) field in the Connect response indicates if NVMe in-band authentication is required. If AUTHREQ is cleared to ~~zero~~ 0h, the created queue is ready for use after the Connect command completes successfully. If AUTHREQ is set to a non-zero value, the created queue is ready for use after NVMe in-band authentication has been performed successfully using the Authentication Send and Authentication Receive Fabrics commands.

…

3.3.3.2 Common Completion Queue Entry

…

Figure 91: Completion Queue Entry: DW 3

Bits	Description
31:17	**Status Field (SF):** Indicates the status for the command that is being completed. Refer to section 3.3.3.2.1.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 91: Completion Queue Entry: DW 3

Bits	Description
16	**Phase Tag (P):** Identifies whether a completion queue entry is new. Refer to section 3.3.3.2.2. ~~The Phase Tag values for all completion queue entries shall be initialized to '0' by host software prior to setting CC.EN to '1'. When the controller places an entry in the Completion Queue, the controller shall invert the Phase Tag to enable host software to discriminate a new entry. Specifically, for the first set of completion queue entries after CC.EN is set to '1' all Phase Tags are set to '1' when they are posted. For the second set of completion queue entries, when the controller has wrapped around to the top of the Completion Queue, all Phase Tags are cleared to '0' when they are posted. The value of the Phase Tag is inverted each pass through the Completion Queue.~~ This is a reserved bit in NVMe over Fabrics implementations.
15:00	**Command Identifier (CID):** Indicates the identifier of the command that is being completed. This identifier is assigned by host software when the command is submitted to the Submission Queue. The combination of the SQ Identifier and Command Identifier uniquely identifies the command that is being completed. The maximum number of requests outstanding for a Submission Queue at one time is 64 Ki.

…

Add a new section 3.3.3.2.2 between "3.3.3.2.1.4 Path Related Status Definition" and "3.3.3.3 Queue Size" as shown below:

3.3.3.2.2 Phase Tag

The Phase Tag bit indicates whether a completion queue entry is new. The Phase Tag bit for each completion queue entry in:

- the Admin Completion Queue shall be initialized to '0' by the host ~~software~~ prior to setting CC.EN (refer to Figure 46) to '1'; and
- an I/O Completion Queue shall be initialized to '0' by the host prior to submitting the Create I/O Completion Queue command for that queue.

When the controller posts a new completion queue entry to the Completion Queue, the controller shall invert the Phase Tag bit in that completion queue entry (i.e., the inverting of the Phase Tag bit enables the host ~~software~~ to detect the new completion queue entry).

When a completion queue entry is posted to a completion queue slot in:

- the Admin Queue for the first time after CC.EN is set to '1', the Phase Tag bit for that completion queue entry is set to '1'; and
- an I/O Completion Queue for the first time after the Create I/O Completion Queue command completed for that queue, the Phase Tag bit for that completion queue entry is set to '1'.

This continues for each completion queue entry that is posted until the controller posts a completion queue entry to the highest numbered completion queue slot and wraps to completion queue slot number 0 as described in section 3.3.1.2. When that queue wrap condition occurs, the Phase Tag bit is then cleared to '0' in each completion queue entry that is posted. This continues until another queue wrap condition occurs. Each time a queue wrap condition occurs, the value of the Phase Tag bit is inverted (i.e., changes from '1' to '0' or changes from '0' to '1').

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

3.3.3.2.2.1 Phase Tag Example

Figure 99a shows an example of how the Phase Tag bit changes over time as the Controller completes commands and the host processes those completions. This example shows a Completion Queue consisting of 6 entries.

Figure 99a: Phase Tag bit Transition Example

T[1]	Condition	Completion Queue Entry/Slot number					
		0	**1**	**2**	**3**	**4**	**5**
0	Admin Queue: Host initializes Completion Queue and sets CC.EN to ‘1’ I/O Queue: Host initializes Completion Queue and submits Create I/O Completion Queue command	P(0) (E) HEAD-> TAIL->	P(0) (E)	P(0) (E)	P(0) (E)	P(0) (E)	P(0) (E)
1	Controller has completed 1 command and the host has consumed 0 completions	P(1) HEAD->	P(0) (E) TAIL->	P(0) (E)	P(0) (E)	P(0) (E)	P(0) (E)
2	Controller has completed 6 commands and the host has consumed 2 completions	P(1) (E) TAIL->	P(1) (E)	P(1) HEAD->	P(1)	P(1)	P(1)
3	Controller has completed 7 commands and the host has consumed 2 completions	P(0)	P(1) (E) TAIL->	P(1) HEAD->	P(1)	P(1)	P(1)
4	Controller has completed 7 commands and the host has consumed 4 completions	P(0)	P(1) (E) TAIL->	P(1) (E)	P(1) (E)	P(1) HEAD->	P(1)
5	Controller has completed 11 commands and the host has consumed 8 completions	P(0) (E)	P(0) (E)	P(0) HEAD->	P(0)	P(0)	P(1) (E) TAIL->
6	Controller has completed 11 commands and the host has consumed 11 completions	P(0) (E)	P(0) (E)	P(0) (E)	P(0) (E)	P(0) (E)	P(1) (E) HEAD-> TAIL->

Key:

P(0) =	Phase Tag bit for this completion queue entry is cleared to the value ‘0’.
P(1) =	Phase Tag bit for this completion queue entry is set to the value ‘1’.
(E) =	The Entry/Slot is empty.
HEAD-> =	Completion Queue Head Pointer for this completion queue is set to indicate this slot.
TAIL-> =	Completion Queue Tail Pointer for this completion queue is used within the controller to indicate this slot.

Note:

[1] T = Time sequence.

At time 0, the host initializes the Completion queue (i.e., clearing the Phase Tag bit to ‘0’ in each completion queue entry in the completion queue). For the Admin Completion Queue, the host then sets CC.EN to ‘1’ to enable the controller. For an I/O Completion Queue, the host then sends the Create I/O Completion Queue command. The queue, at this time, is in the Empty condition (refer to section 3.3.1.4).

At time 1, the controller has completed a command, but the host has not consumed that completion queue entry. As a result of the command completion, the Phase Tag bit in completion queue entry 0 has been inverted to ‘1’. Since no completion queue entries have been consumed, the Completion Queue Head pointer still indicates completion queue entry 0. The controller has updated the internal Completion Queue

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Tail Pointer to indicate that completion queue slot 1 is the next completion queue slot into which the controller posts a completion queue entry.

At time 2, the controller has completed 5 additional commands (i.e., 6 commands have been completed) and the host has consumed 2 of the completion queue entries. As a result of the 5 additional commands having been completed, the Phase Tag bit has been inverted to ‘1’ in completion queue entry 1 through completion queue entry 5. As a result of 2 completion queue entries having been consumed, the host has updated the Completion Queue Head Pointer to indicate that completion queue entry 0 and completion queue entry 1 have been consumed (i.e., completion queue entry 2 is the next completion queue entry for the host to consume). The controller has updated the internal Completion Queue Tail Pointer to indicate that completion queue slot 0 is the next completion queue slot into which the controller posts a completion queue entry.

At time 3, the controller has completed 1 additional command (i.e., 7 commands have been completed) and no additional completion queue entries have been consumed by the host (i.e., 2 completion queue entries have been consumed). As a result of the additional command having been completed, the Phase Tag bit has been inverted to ‘0’ in completion queue entry 0 (i.e., accounting for the queue wrap condition). The controller has updated the internal Completion Queue Tail Pointer to indicate that completion queue slot 1 is the next completion queue slot into which the controller posts a completion queue entry. The queue, at this time, is in the Full condition (refer to section 3.3.1.5).

At time 4, the controller has completed no additional commands (i.e., 7 commands have been completed) and the host has consumed 2 additional completion queue entries (i.e., 4 completion queue entries have been consumed). As a result of 2 additional completion queue entries having been consumed, the host has updated the Completion Queue Head Pointer to indicate that completion queue entry 2 and completion queue entry 3 have now been consumed (i.e., completion queue entry 4 is the next completion queue entry for the host to consume). The controller internal Completion Queue Tail Pointer has not changed.

At time 5, the controller has completed 11 commands and the host has consumed 8 of the completion queue entries. As a result of the 4 additional commands having been completed, the Phase Tag bit has been inverted to ‘0’ in completion queue entry 1 through completion queue entry 4. As a result of the 4 additional completion queue entries having been consumed, the host has updated the Completion Queue Head Pointer to indicate that completion queue entry 5 through completion queue entry 1 (i.e., accounting for the queue wrap condition) have now been consumed (i.e., completion queue entry 2 is the next completion queue entry for the host to consume). The controller has updated the internal Completion Queue Tail Pointer to indicate that completion queue slot 5 is the next completion queue slot into which the controller posts a completion queue entry.

At time 6, the controller has completed 11 commands and the host has consumed all 11 of the completion queue entries. As a result of no new command completions, there are no changes to the Phase Tag bit values. As a result of the 3 additional completion queue entries having been consumed, the host has updated the Completion Queue Head Pointer to indicate that completion queue entry 2 through completion queue entry 4 have now been consumed (i.e., completion queue slot 5 is the next completion queue slot from which the host consumes a completion queue entry). The queue, at this time, is in the Empty condition (refer to section 3.3.1.4).
…

3.4.1 Command Ordering Requirements

Commands which are not part of a fused operation (refer to section 3.4.2) and which comply with atomic operations requirements (refer to section 3.4.3), are processed as independent entities without reference to other commands submitted to the same I/O Submission Queue or to commands submitted to other I/O Submission Queues. For example, the controller is not responsible for checking the LBA of a NVM Command Set Read command or Write command to ensure any type of ordering between commands. If a Read command is submitted for LBA *x* and there is a Write command also submitted for LBA *x*, there is no guarantee of the order of completion for those commands (the Read command may finish first or the Write command may finish first). If there are ordering requirements between these commands, host software or the associated application is required to enforce that ordering above the level of the controller.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

…

Modify a portion of section 3.5 as shown below:

3.5 Controller Initialization

…

3.5.2 Message-based Transport Controller Initialization

…

<Editors note: The reference to step 5 below should be made a hot link so a future change to that list doesn't make this reference wrong.>

The association may be removed if step ~~4~~5 (i.e., setting CC.EN to '1') is not completed within 2 minutes after establishing the association.

3.5.3 Controller Ready Modes During Initialization

There are two ~~possible~~ controller ready modes:

…

Controllers ~~are required to~~ shall set the CAP.CRMS.CRWMS bit to '1' (i.e., ~~to~~ set the CAP.CRMS field to 01b or 11b). The CAP.CRMS.CRWMS bit was not defined ~~in~~ prior to NVM Express Base Specification, R~~r~~evision ~~1.4 and earlier~~ 2.0. Controllers compliant with revisions earlier than NVM Express Base Specification, R~~r~~evision ~~1.4~~ 2.0 ~~and earlier~~ may ~~have~~ clear the CAP.CRMS.CRWMS field ~~cleared~~ to 00b.

…

3.5.4 Controller Ready Timeouts During Initialization

The CAP.CRMS field was not defined ~~in~~ prior to NVM Express Base Specification, R~~r~~evision ~~1.4 and earlier~~ 2.0. Controllers compliant with revisions earlier than NVM Express Base Specification, R~~r~~evision ~~1.4~~ 2.0 ~~and earlier~~ may clear the CAP.CRMS field to 00b. This section is applicable to controllers that clear the CAP.CRMS field to 00b and controllers that set CAP.CRMS to a non-zero value.

…

Modify a portion of section 3.7 as shown below:

3.7.1.1 Single Domain NVM Subsystems

…

An NVM Subsystem Reset is initiated when:

- Main power is applied to the NVM subsystem;
- A value of 4E564D65h ("NVMe") is written to the NSSR.NSSRC field;
- Requested using a method defined in the NVM~~e~~ Express Management Interface ~~s~~Specification; or

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

- A vendor specific event occurs.

…

Modify a portion of section 3.8 as shown below:

3.8 NVM Capacity Model

…

3.8.2.3 Horizontally-Organized Dual NAND NVM Subsystem

…

The Capacity Configuration Descriptor, ~~would~~ for this example, contains two Endurance Group Configuration Descriptors. The first Endurance Group Configuration Descriptor for this example:

- indicates a Capacity Adjustment Factor of approximately ~~1,600~~ 400;
- contains one NVM Set Identifier; and
- contains four Channel Configuration Descriptors. Each Channel Configuration Descriptor contains one Media Unit Configuration Descriptor.

…

Modify a portion of section 3.9 as shown below:

3.9 Keep Alive

…

The Keep Alive timer is active if:

- CC.EN is set to ‘1’;
- CSTS.RDY is set to ‘1’;
- CC.SHN is cleared to ‘00b’;
- CSTS.SHST is cleared to ‘00b’; and
- the Keep Alive Timer feature has been enabled with a KATO field (refer to section 5.27.1.12 and section 6.3 ~~or the Fabric Connect command in the NVMe over Fabrics specification~~) set to a non-zero value,

…

Modify a portion of section 3.11 as shown below:

3.11 Firmware Update Process

…

Host software should use the same controller or Management Endpoint (refer to the NVM~~e~~ Express Management Interface ~~s~~Specification) for all commands that are part of a firmware image update command sequence. If the commands for a single firmware/boot partition image update command sequence are

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

submitted to more than one controller and/or Management Endpoint, the controller may abort the Firmware Commit command with Invalid Firmware Image status.

…

Modify a portion of section 4.1 as shown below:

4.1 Data Layout

…

4.1.2 Scatter Gather List (SGL)

A Scatter Gather List (SGL) is a data structure in memory address space used to describe a data buffer. The controller indicates the SGL types that the controller supports in the Identify Controller data structure. A data buffer is either a source buffer or a destination buffer. An SGL contains one or more SGL segments. The total length of the Data Block and Bit Bucket descriptors in an SGL shall be equal to or exceed the amount of data requested to be transferred. If the amount of data requested to be transferred exceeds the total length of the Data Block and Bit Bucket descriptors in an SGL, data shall not be transferred to or from locations that are not described by the SGL.

…

The value of the SDT field shall be less than or equal to the value of the Maximum SGL Data Block Descriptors field (MSDBD) field in the Identify Controller data structure (refer to ~~the NVMe over Fabrics specification~~ Figure 275 for the definition of the MSDBD field).

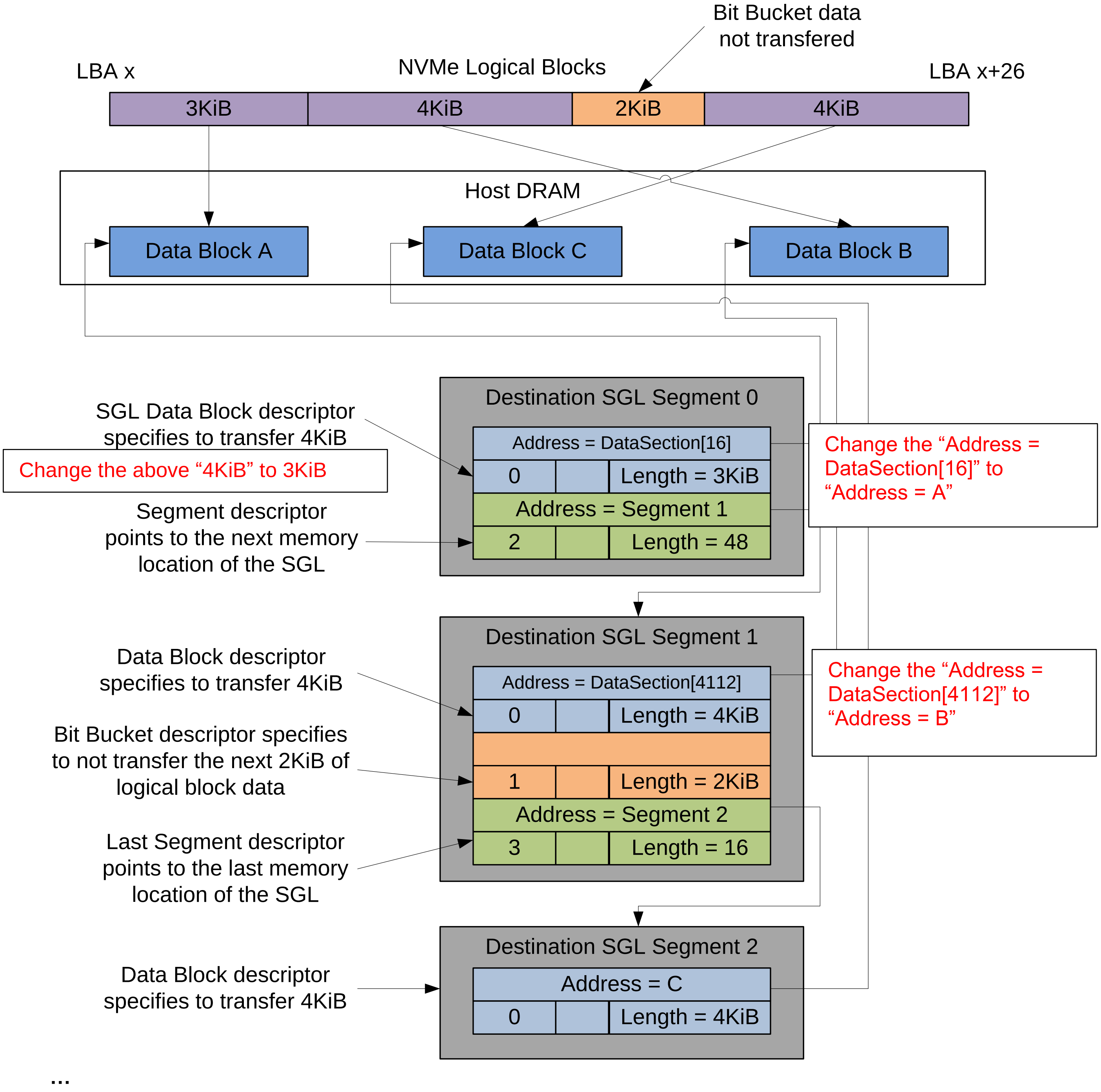
…

4.1.2.1 SGL Example

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 122: SGL Read Example


Bit Bucket data not transfered
LBA x
NVMe Logical Blocks
LBA x+26
3KiB
4KiB
2KiB
4KiB
Host DRAM
Data Block A
Data Block C
Data Block B
SGL Data Block descriptor specifies to transfer 4KiB
Change the above “4KiB” to 3KiB
Destination SGL Segment 0
Address = DataSection[16]
0
Length = 3KiB
Address = Segment 1
2
Length = 48
Change the “Address = DataSection[16]” to “Address = A”
Segment descriptor points to the next memory location of the SGL
Destination SGL Segment 1
Data Block descriptor specifies to transfer 4KiB
Address = DataSection[4112]
0
Length = 4KiB
Change the “Address = DataSection[4112]” to “Address = B”
Bit Bucket descriptor specifies to not transfer the next 2KiB of logical block data
1
Length = 2KiB
Address = Segment 2
Last Segment descriptor points to the last memory location of the SGL
3
Length = 16
Destination SGL Segment 2
Data Block descriptor specifies to transfer 4KiB
Address = C
0
Length = 4KiB
…

Modify a portion of section 5 as shown below:

5 Admin Command Set

…

Figure 139: Sanitize Operations and Format NVM Command – Admin Commands Allowed

<table>
<tr><th>Admin Command</th><th>Additional Restrictions for Format NVM command</th><th>Additional Restrictions for sanitize operations</th></tr>
<tr><td>…</td><td colspan="2"></td></tr>
<tr><td>Opcode 7Fh</td><td colspan="2">The Fabric Commands allowed are listed below. Refer to ~~the NVMe over Fabrics specification~~ section 6.
<table>
<tr><th>Fabrics Commands</th><th>Additional Restrictions for both Format NVM command and sanitize operations</th></tr>
<tr><td>Property Set</td><td></td></tr>
<tr><td>Connect</td><td></td></tr>
<tr><td>Disconnect</td><td></td></tr>
<tr><td>Property Get</td><td></td></tr>
<tr><td>Authentication Send</td><td></td></tr>
<tr><td>Authentication Receive</td><td></td></tr>
<tr><td>Vendor Specific</td><td>Commands are allowed that do not affect or retrieve user data.</td></tr>
</table>
</td></tr>
<tr><td>…</td><td colspan="2"></td></tr>
</table>

…

Modify a portion of section 5.2 as shown below:

5.2 Asynchronous Event Request command

…

5.2.1 Command Completion

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 146: Asynchronous Event Information – Notice

Value	Description
00h	**Namespace Attribute Changed:** Indicates a change to one or more of the following: • the Identify Namespace data structure (refer to the applicable NVMe I/O Command Set specification) for one or more namespaces; ~~or~~ • the I/O Command Set Independent Identify Namespace data structure; ~~or~~ • the Namespace List returned when the Identify command is issued with the CNS field set to 02h; or • other data structures as specified in applicable NVMe I/O Command Set specifications. To clear this event, host software issues a Get Log Page command for the Changed Namespace List log page (refer to section 5.16.1.5) with the Retain Asynchronous Event bit cleared to ‘0’. A controller shall not send this event if: a) Namespace Status (refer to Figure 280) has changed, and shutdown processing is either occurring (i.e., CSTS.SHST is set to 01b) or complete (i.e., CSTS.SHST is set to 10b); ~~or~~ b) the ANAGRPID field (refer to Figure 280) has changed; or c) an I/O Command Set specific change occurs (refer to the applicable I/O Command Set specification).~~; or~~ ~~d) Namespace Status (refer to Figure 280) has changed and shutdown processing is either occurring (i.e., CSTS.SHST is set to 01b) or complete (i.e., CSTS.SHST is set to 10b).~~ A controller shall only send this event for changes to the Format Progress Indicator field when bits 6:0 of that field transition from a non-zero value to 0h, or from 0h to a non-zero value.
…	
03h	**Asymmetric Namespace Access Change:** The Asymmetric Namespace Access information (refer to section 5.16.1.13) related to an ANA Group that contains namespaces attached to this controller has changed (e.g., an ANA state has changed, an ANAGRPID has changed). The current Asymmetric Namespace Access information for attached namespaces is indicated in the Asymmetric Namespace Access log page (refer to section 5.16.1.13). To clear this event, the host issues a Get Log Page command (refer to section 5.16) with the Retain Asynchronous Event bit cleared to ‘0’ for the Asymmetric Namespace Access log. A controller shall not send this event if a Namespace Attribute Changed notice is sent for the same event, such as a change due to: a) the attachment of a namespace (refer to section 5.22); b) the deletion of a namespace (refer to section 5.23); or c) the detachment of a namespace (refer to section 5.22).
…	

…

Modify a portion of section 5.10 as shown below:

5.10 Directive Receive command

…

5.10.1 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 8.7.

~~Directive Receive command specific status values are defined in Figure 177.~~

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

~~**Figure 177: 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 8.12).~~

Modify a portion of section 5.12 as shown below:

5.12 Firmware Commit command

…

5.12.1 Command Completion

…

Firmware Commit command specific status values (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 183.

Figure 183: Firmware Commit – Command Specific Status Values

Value	Description
…	
07h	**Invalid Firmware Image:** The firmware image specified for activation is: • invalid and not loaded by the controller; or • the specified firmware slot does not contain a firmware image.

…

Modify a portion of section 5.14 as shown below:

5.14 Format NVM command

…

If a host does not set the LBA Format Extension Enable (LBAFEE) field to 1h in the Host Behavior Support feature (refer to section 5.27.1.18), then ~~a controller aborts a Format NVM command that specifies I/O Command Set specific formats as specified in the applicable I/O Command Set specification which require the LBAFEE field to be set to 1h with a status code of Invalid Namespace or Format.~~ the 0h value of the LBAFEE field disables any I/O Command Set specific format that requires the LBAFEE field to be set to 1h (refer to the applicable I/O Command Set specification). If a Format NVM command specifies a format that is disabled (e.g., the LBAFEE field is cleared to 0h), then the controller shall abort that Format NVM command with a status code of Invalid Namespace or Format.

If the format operation scope (refer to Figure 188) for a Format NVM command includes any namespace that is write protected (refer to section 8.12), then the controller aborts that Format NVM command with a status code of Namespace is Write Protected.

If bit 3 in the FNA field is set to ‘1’ and a Format NVM command has the NSID field set to FFFFFFFFh, then the controller shall abort the command with a status code of Invalid Field In Command.

…

5.14.1 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 (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 190.

Figure 190: 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 User Data Format number; 2. enabling protection information when there are not sufficient metadata resources; or 3. the specified format is not available in the current configuration.

…

Modify a portion of section 5.15 as shown below:

5.15 Get Features command

The Get Features command retrieves the attributes of the Feature specified.

The Get Features command uses the Data Pointer, Command Dword 10 and Command Dword 14 fields. The use of the Command Dword 11 field 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.

The mandatory, optional, and prohibited Feature Identifiers for each type of controller are defined in section 3.1.2.1.3, section 3.1.2.2.3, and section 3.1.2.3.4.

…

5.15.2 Command Completion

Upon completion of the Get Features command, the controller posts a completion queue entry to the Admin Completion Queue. If the Select field is not set to 011b, then Dword 0 of the completion queue entry may contain feature-dependent information (refer to section 5.27.1).

If the Select field is set to 011b, then Figure 195 describes the contents of Dword 0 of the completion queue entry.

…

Modify a portion of section 5.16 as shown below:

5.16 Get Log Page command

…

Figure 197: Get Log Page – Command Dword 10

Bits	Description
…	
15	**Retain Asynchronous Event (RAE):** This bit specifies ~~when~~ whether to retain or clear an Asynchronous Event. If this bit is cleared to '0', the corresponding Asynchronous Event is cleared ~~after the~~ by the controller upon successful command completion~~es successfully~~. If this bit is set to '1', the corresponding Asynchronous Event is retained (i.e., not cleared) ~~after the~~ by the controller upon command completion~~es successfully~~. If the command does not complete successfully, the Asynchronous Event shall be retained by the controller. Host software should clear this bit to '0' for log pages that are not used with Asynchronous Events. Refer to section 5.2.
…	

…

5.16.1.1 Supported Log Pages (Log Identifier 00h)

…

Figure 204: Get Log Page – LID Supported and Effects Data Structure

Bits	Description
31:16	**LID Specific Field:** This field is specific to the log page identifier as defined in Figure 205.
~~31~~15:2	Reserved
…	

Figure 205: LID Supported and Effects Data Structure – LID Specific Field

<table>
<tr><th>Log Page Identifier</th><th>LID Specific Field</th></tr>
<tr><td>0 to Ch</td><td>Reserved</td></tr>
<tr><td>0Dh</td><td>The LID Specific Field for log page identifier 0Dh (Persistent Event Log as described in section 5.16.1.14) is defined as follows:
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>15:1</td><td>Reserved</td></tr>
<tr><td>0</td><td><b>Establish Context and Read 512 Bytes of Header Supported:</b> If this bit is cleared to '0', then the controller does not support the Establish Context and Read 512 Bytes of Header action (refer to Figure 223).
If this bit is set to '1', then the controller supports:
• the Establish Context and Read 512 Bytes of Header action; and~~. If this bit is set to '1', then~~
• the Generation Number field in the Persistent Event Log ~~shall also be supported~~.
Implementations compliant with ~~to later than~~ NVM Express Base Specification, R~~r~~evision ~~1.4~~ 2.0 and later shall set this bit to '1'.</td></tr>
</table></td></tr>
<tr><td>0Eh to BFh</td><td>Reserved</td></tr>
<tr><td>C0h to FFh</td><td>Vendor specific</td></tr>
</table>

…

5.16.1.2 Error Information (Log Identifier 01h)

…

Figure 206: Get Log Page – Error Information Log Entry (Log Identifier 01h)

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>15:14</td><td><b>Parameter Error Location:</b> This field indicates the byte and bit of the command parameter that the error is associated with, if applicable. If the parameter spans multiple bytes or bits, then the location indicates the ~~first~~ least-significant byte and bit of the parameter.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>15:11</td><td>Reserved</td></tr>
<tr><td>10:08</td><td><b>Bit Location:</b> The offset in the byte specified by the Byte Location field to the ~~B~~bit in that byte ~~command~~ that contained the error. ~~Valid values are 0 to 7.~~</td></tr>
<tr><td>07:00</td><td><b>Byte Location:</b> The offset in the submission queue entry to the ~~B~~byte in the command that contained the error. Valid values are based on the SQES field (refer to Figure 275) (e.g., a value of 6 in SQES indicates that the valid values for this field are 0 to 63).</td></tr>
</table>

If the error is not specific to a particular command, then this field shall be set to FFFFh.</td></tr>
<tr><td>…</td><td></td></tr>
</table>

…

5.16.1.3 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 unless otherwise specified. To request the controller log page, the namespace identifier specified is FFFFFFFFh or 0h. For compatibility with implementations compliant with NVM Express Base Specification, R~~r~~evision 1.4 and earlier, hosts should use a namespace identifier of FFFFFFFFh to request the controller log page. The controller may also support requesting the log page on a per namespace basis, as indicated by bit 0 of the LPA field in the Identify Controller data structure in Figure 275.

…

Figure 207: Get Log Page – SMART / Health Information Log

Bytes	Description
00	**Critical Warning:** This field indicates critical warnings for the state of the controller. Each bit corresponds to a critical warning type; multiple bits may be set to '1'. If a bit is cleared to '0', then that critical warning does not apply. Critical warnings may result in an asynchronous event notification to the host. Bits in this field represent the ~~current associated~~ state at the time the Get Log Page command is processed and may not reflect the state at the time a related asynchronous event notification, if any, occurs or occurred ~~and are not persistent~~. …
…	

…

5.16.1.5 Changed Namespace List (Log Identifier 04h)

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

a) changed information in their Identify Namespace data structures (refer to in Figure 146) since the last time the log page was read;

b) changed information in their I/O Command Set Independent Identify Namespace data structure since the last time the log page was read;

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

c) been added; and
d) been deleted.

…

5.16.1.9 Telemetry Controller-Initiated (Log Identifier 08h)

…

The controller shall return data for all blocks requested:

- If bit 6 of the Log Page Attributes field is cleared to ‘0’, then the data beyond the last block in Telemetry Controller-Initiated Data Area 3 Last Block is undefined.
- If bit 6 of the Log Page Attributes field is set to ‘1’, then the data beyond the last block in Telemetry Controller-Initiated Data Area 4 Last Block is undefined. ~~If the host requests a data transfer that is not a multiple of 512 bytes, then the controller shall return an error of Invalid Field in Command.~~
- If bit 6 of the Log Page Attributes field is set to ‘1’ and the Extended Telemetry Data Area 4 Supported (ETDAS) field is cleared to 0h in the Host Behavior Support feature, then the data beyond the last block in Telemetry Controller-Initiated Data Area 3 Last Block is undefined.

If the host requests a data transfer that is not a multiple of 512 bytes, then the controller shall abort the command with the status code of Invalid Field in Command.

…

5.16.1.10 Endurance Group Information (Log Identifier 09h)

…

Figure 217: Get Log Page – Endurance Group Information Log (Log Identifier 09h)

Bytes	Description
00	**Critical Warning:** This field indicates critical warnings for the state of the Endurance Group. Each bit corresponds to a critical warning type; multiple bits may be set to ‘1’. If a bit is cleared to ‘0’, then that critical warning does not apply. Critical warnings may result in an asynchronous event notification to the host. Bits in this field represent the ~~current associated~~ state at the time the Get Log Page command is processed and may not reflect the state at the time a related asynchronous event notification, if any, occurs or occurred ~~and are not persistent~~. If a bit is set to ‘1’ in all Endurance Groups in the NVM subsystem, then the corresponding bit shall be set to ‘1’ in the Critical Warning field of the SMART / Health Information log page (refer to Figure 207). …
…	

…

5.16.1.13 Asymmetric Namespace Access (Log Identifier 0Ch)

…

Figure 221: Get Log Page – Asymmetric Namespace Access Log

Bytes	Description
…	

Figure 221: Get Log Page – Asymmetric Namespace Access Log

Bytes	Description
09:08	**Number of ANA Group Descriptors:** This field indicates the number of ANA Group Descriptors available in the log page. The log page shall contain one ANA Group Descriptor for each ANA Group that contains ~~N~~namespaces that are attached to the controller. If, for an ANA Group, there are no namespaces attached to the controller processing the command, then no ANA Group Descriptor is returned for that ANA Group (i.e., an ANA Group Descriptor is returned only if that ANA Group contains namespaces that are attached to the controller processing the command). If no namespaces are attached to the controller, then the log page does not contain any ANA Group Descriptors and this field is cleared to 0h.
15:10	Reserved
n:16	**ANA Group Descriptor 0, if any**
m:n+1	**ANA Group Descriptor 1, if any**
…	**…**
x:y	**ANA Group Descriptor n, if any**

The format of the ANA Group Descriptor is defined in Figure 222. Namespace Identifiers shall be listed in ascending NSID order.

…

5.16.1.14 Persistent Event Log (Log Identifier 0Dh)

…

A sanitize operation may alter this log page (e.g., remove or modify events to prevent derivation of user data from log page information, refer to section ~~8.20~~ 8.21). The events removed from this log page by a sanitize operation are unspecified.

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 224: Get Log Page – Persistent Event Log (Log Identifier 0Dh)

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>377:374</td><td>

Reporting Context Information (RCI): This field contains information about the persistent event log reporting context.

<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>31:19</td><td>Reserved</td></tr>
<tr><td>18</td><td><b>Reporting Context Exists (RCE):</b> This bit indicates the persistent event log reporting context. If this bit is set to ‘1’, then a persistent event log reporting context already existed when the Get Log Page command that requested this log page was processed. If this bit is cleared to ‘0’, then a persistent event log reporting context did not already exist when the Get Log Page command that requested this log page was processed.</td></tr>
<tr><td>17:16</td><td>

<b>Reporting Context Port Identifier Type (RCPIT):</b> If the RCE bit is set to ‘1’, then this field indicates the type of port identifier reported in the Reporting Context Port Identifier (RCPID) field. If the RCE bit is cleared to ‘0’, then this field shall be cleared to 00b.

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>A persistent event log reporting context does not already exist.</td></tr>
<tr><td>01b</td><td>The reporting context was established by an NVM subsystem port ~~and the Reporting Context Port Identifier (RCPID) field indicates the Port Identifier of the NVM subsystem port that established the reporting context~~.</td></tr>
<tr><td>10b</td><td>The reporting context was established by a Management Endpoint (refer to the NVM Express Management Interface Specification)~~an NVMe-MI port and the Reporting Context Port Identifier (RCPID) field indicates the Port Identifier of the NVMe-MI port that established the reporting context~~.</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table>

</td></tr>
<tr><td>15:00</td><td>

<b>Reporting Context Port Identifier (RCPID):</b> If the RCE bit is cleared to ‘0’, then this field shall be cleared to 0h.

If the RCE bit is set to ‘1’, then this field contains a Port Identifier of the type indicated in the RCPIT field.

If the RCPIT field is set to 01b, then this field shall contain the Port Identifier of the NVM subsystem port that established the reporting context as defined in the Primary Controller Capabilities data structure (refer to section 5.17.2.13).

If the RCPIT field is set to 10b, then:

- the least-significant byte of this field shall contain the Port Identifier of the Management Endpoint ~~as defined in the Controller Information Data Structure~~ (refer to the NVM Express Management Interface Specification); and
- the most-significant byte of this field shall be cleared to 0h.

</td></tr>
</table>

</td></tr>
<tr><td>…</td><td></td></tr>
</table>

…

Figure 225: Persistent Event Format

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td colspan="2"><b>Persistent Event Log Event Header</b></td></tr>
<tr><td>…</td><td></td></tr>
</table>

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 225: Persistent Event Format

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>03</td><td><b>Event Header Additional Information (EHAI):</b> This field indicates if additional information is present in this event header.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>7:2</td><td>Reserved</td></tr>
<tr><td>1:0</td><td><b>Port Identifier Type (PIT):</b> This field indicates the type of port identifier reported in the Port Identifier (PELPID) field. Implementations that are compliant with NVM Express Base Specification revision 1.4 and later shall not clear this field to 0h.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>The Port Identifier Type is not reported and the Port Identifier (PELPID) field does not apply.</td></tr>
<tr><td>01b</td><td>This event is associated with an NVM subsystem port ~~and the Port Identifier (PELPID) field indicates the Port Identifier of the NVM subsystem port associated with this event~~.</td></tr>
<tr><td>10b</td><td>This event is associated with a Management Endpoint (refer to the NVM Express Management Interface Specification)~~an NVMe-MI port and the Port Identifier (PELPID) field indicates the Port Identifier of the NVMe-MI port associated with this event~~.</td></tr>
<tr><td>11b</td><td>This event is not associated with any port. ~~and the Port Identifier (PELPID) field does not apply~~</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>15:14</td><td><b>Port Identifier (PELPID):</b> If the PIT field in the EHAI field is cleared to 00b or set to 11b, then this field shall be cleared to 0h.
If the PIT field in the EHAI field is not cleared to 00b or set to 11b, then this field contains a Port Identifier of the type indicated in the PIT field.
If the PIT field in the EHAI field is set to ~~1h~~ 01b, then this field shall contain the Port Identifier of the NVM subsystem port associated with this event as defined in the Primary Controller Capabilities data structure (refer to section 5.17.2.13).
If the PIT field in the EHAI field is set to 10b, then:
• the least-significant byte of this field shall contain the Port Identifier of the Management Endpoint associated with this event ~~as defined in the Controller Information Data Structure~~ (refer to the NVM Express Management Interface Specification); and
• the most-significant byte of this field shall be cleared to 0h.</td></tr>
<tr><td>…</td><td></td></tr>
</table>

…

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

…

Figure 233: NVM Subsystem Hardware Error Event Codes

Code	Description
…	

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 233: NVM Subsystem Hardware Error Event Codes

<table>
<tr><th>Code</th><th>Description</th></tr>
<tr><td>07h</td><td><b>Endurance Group Critical Warning Condition:</b> Indicates that the NVM subsystem has detected a condition that causes a bit in the Critical Warning field of an Endurance Group Information log page (refer to section 5.16.1.10) to be set to ‘1’.
Bits in this field represent the ~~associated~~ state at the time ~~of~~ this event is added to the Persistent Event log page.
The Additional Hardware Error Information field shall be four bytes long and contain the following information:~~.~~
<table>
<tr><th>Bytes</th><th>Definition</th></tr>
<tr><td>0</td><td>Shall be set at the time ~~of the~~ this event is added to the Persistent Event log page using the same format as is specified for the Critical Warning field of the Endurance Group Information log page.</td></tr>
<tr><td>1</td><td>Reserved</td></tr>
<tr><td>3:2</td><td>Shall be set to the Endurance Group Identifier for the associated ~~e~~Endurance ~~g~~Group.</td></tr>
</table></td></tr>
<tr><td>…</td><td></td></tr>
</table>

…

5.16.1.14.1.8 Format NVM Completion Event (Event Type 08h)

A Format NVM Completion event shall be recorded in the Persistent Event Log at the completion of a Format NVM command that resulted in modification of the contents of the NVM.

The Format NVM Completion event shall set the Persistent Event Log Event Format Header:

- Event Type field to 08h; and
- Event Type Revision field to ~~01h~~ 02h.

Figure 238: Format NVM Completion Event Data Format (Event Type 08h)

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>~~11~~09:08</td><td>~~<b>Status Field:</b> 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.~~
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>15:1</td><td><b>Status:</b> This field indicates the value that was reported in the Status 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.</td></tr>
<tr><td>0</td><td><b>Phase Tag:</b> This field may indicate the Phase Tag posted for the command.</td></tr>
</table></td></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
</table>

…

5.16.1.14.1.11 Set Feature Event (Event Type 0Bh)

…

Figure 241: Set Feature Event Data Format

Bytes	Description
…	

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 241: Set Feature Event Data Format

Bytes	Description
(Dword Count * 4) + 3: 4	**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 the Common Command Format in Figure 87.
~~Data~~ (Memory Buffer Count) + (Dword Count * 4) + 34: (Dword Count * 4) + 4	**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.
~~Data~~ Memory Buffer Count + (Dword Count * 4) + ~~8~~ 7: ~~Data~~ Memory Buffer Count + (Dword Count * 4) + ~~5~~ 4	**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.

…

5.16.1.14.1.12 Telemetry Log Create Event (Event Type 0Ch)

A Telemetry Log Create event may be created if the controller determines that a Telemetry ~~h~~Host-~~i~~Initiated ~~telemetry~~ log page (refer to section 5.16.1.8) or that a Telemetry ~~c~~Controller-~~i~~Initiated ~~telemetry~~ log page (refer to section 5.16.1.9) has been generated.

The Telemetry Log Create Event shall set the Persistent Event Log Event Format Header:

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

Figure 242: Telemetry Log Create Event Data Format (Event Type 0Ch)

Bytes	Description
511:00	**Telemetry Initiated Log:** Contains a copy of the values from the first 512 bytes of the Telemetry Host-Initiated ~~L~~log page (refer to Figure 215) or the first 512 bytes of the Telemetry Controller-Initiated ~~L~~log page (refer to~~ Figure 215 and~~ Figure 216).

…

5.16.1.17 Supported Capacity Configuration List (Log Identifier 11h)

…

In the Supported Capacity Configuration List (refer to Figure 250), Capacity Configuration Descriptors ~~are~~ shall be listed in ascending order by Capacity Configuration Identifier, and each Capacity Configuration Identifier shall appear only once.

…

In the Capacity Configuration Descriptor (refer to Figure 251), Endurance Group Configuration Descriptors ~~are~~ shall be listed in ascending order by Endurance Group Identifier, and each Endurance Group Identifier shall appear only once.

…

In the Endurance Group Configuration Descriptor (refer to Figure 252), NVM Set Identifiers ~~are~~ shall be listed in ascending order by value, and each NVM Set Identifier shall appear only once.

In the Endurance Group Configuration Descriptor, Channel Configuration Descriptors ~~are~~ shall be listed in ascending order by Channel Identifier value, and each Channel Identifier shall appear only once.

<Editors note: Correct the table heading line to make it the first row of the table – so it propagates correctly onto subsequent pages.>

Figure 252: Endurance Group Configuration Descriptor

Bytes	Description
…	
NVM Set Identifiers	
81:80	**Number of NVM Sets (EGSETS):** This field indicates the number of NVM Set Identifiers in this Endurance Group Configuration Descriptor. A value of 0h indicates that no NVM Set Identifiers are reported for this Endurance Group.
83:82	**NVM Set 0 Identifier:** This field indicates the identifier of the first NVM Set assigned to this Endurance Group, if reported. Refer to section 3.2.2. ~~This field shall indicate a value greater than or equal to 0h and less than or equal to the value of the NVM Set Identifier Maximum field in the Identify Controller data structure.~~
85:84	**NVM Set 1 Identifier:** This field indicates the identifier of the second NVM Set assigned to this Endurance Group, if reported. ~~This field shall indicate a value greater than or equal to 0h and less than or equal to the value of the NVM Set Identifier Maximum field in the Identify Controller data structure.~~
	…
(EGSETS*2)+81 : (EGSETS*2)+80	**NVM Set EGSETS-1 Identifier:** This field indicates the identifier of the last NVM Set assigned to this Endurance Group, if reported. ~~This field shall indicate a value greater than or equal to 0h and less than or equal to the value of the NVM Set Identifier Maximum field in the Identify Controller data structure.~~
Channel Configuration Descriptors	
…	
Notes: 1. Channel Configuration Descriptors may be different lengths.	

The Channel Configuration Descriptor (refer to Figure 253) lists the Media Units attached to a Channel. Media Unit Configuration Descriptors (refer to Figure 254) ~~are~~ shall be listed in ascending order by Media Unit Identifier, and each Media Unit Identifier shall appear only once.

…

Modify a portion of section 5.17 as shown below:

5.17 Identify command

5.17.1 Identify command overview

…

Note: The CNS field was specified as a one bit field in revision 1.0 and ~~as~~ is a two bit field in revision 1.1. Host software should only issue CNS values defined in revision 1.0 to controllers compliant with revision 1.0. Host software should only issue CNS values defined in revision 1.1 to controllers compliant with revision 1.1. The results of issuing other CNS values to controllers compliant with revision 1.0 or revision 1.1, respectively, are indeterminate.

…

Figure 273: Identify – CNS Values

CNS Value	O/M[1]	Definition	NSID[2]	CNTID[3]	CSI[4]	Reference Section
…						

Figure 273: Identify – CNS Values

CNS Value	O/M[1]	Definition	NSID[2]	CNTID[3]	CSI[4]	Reference Section
NOTES: … 8. This Identify data structure applies to namespaces that are associated with command sets that specify logical blocks (~~i.e.~~e.g., Command Set Identifier 0h or Command Set Identifier 2h). …						

…

5.17.2.1 Identify Controller data structure (CNS 01h)

…

<Editors note: Correct the footnote font size throughout this table <14 point>.>

Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O[1]	Admin[1]	Disc[1]	Description
…				
83:80	M	M	M	**Version (VER):** This field contains the value reported in the Version property (i.e., VS property) defined in section 3.1.3.2. Implementations compliant ~~to~~ with NVM Express Base Specification, R~~r~~evision 1.2 or later shall report a non-zero value in this field.
…				
111	M	M	M	**Controller Type (CNTRLTYPE):** This field specifies the controller type. A value of 0h indicates that the controller type is not reported. Implementations compliant ~~to~~ with NVM Express Base Specification, R~~r~~evision 1.4 or later shall report a controller type (i.e., the value 0h is reserved and shall not be used). Implementations compliant ~~to~~ with an earlier specification version may report a value of 0h to indicate that a controller type is not reported. …
…				
253	M	M	M	**NVM Subsystem Report (NVMSR):** This field reports information associated with the NVM subsystem. If the controller is compliant ~~to~~ with the NVM~~e~~ Express Management Interface Specification, then at least one bit in this field is set to '1'. If the NVM subsystem does not support the NVM~~e~~ Express Management Interface Specification, then this field shall be cleared to 0h. Refer to the NVM~~e~~ Express Management Interface Specification. …
…				
267:266	M	M	R	**Warning Composite Temperature Threshold (WCTEMP):** This field indicates the minimum Composite Temperature field value (reported in the SMART / Health Information log in Figure 207) that indicates an overheating condition during which controller operation continues. Immediate remediation is recommended (e.g., additional cooling or workload reduction). The platform should strive to maintain a composite temperature less than this value. A value of 0h in this field indicates that no warning temperature threshold value is reported by the controller. Implementations compliant ~~to~~ with NVM Express Base Specification, R~~r~~evision 1.2 or later shall report a non-zero value in this field. It is recommended that implementations report a value of 0157h in this field.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 275: Identify – Identify Controller Data Structure, I/O Command Set Independent

<table>
<tr><th>Bytes</th><th>I/O[1]</th><th>Admin[1]</th><th>Disc[1]</th><th>Description</th></tr>
<tr><td>269:268</td><td>M</td><td>M</td><td>R</td><td><b>Critical Composite Temperature Threshold (CCTEMP):</b> This field indicates the minimum Composite Temperature field value (reported in the SMART / Health Information log in Figure 207) that indicates a critical overheating condition (e.g., may prevent continued normal operation, possibility of data loss, automatic device shutdown, extreme performance throttling, or permanent damage).
A value of 0h in this field indicates that no critical temperature threshold value is reported by the controller. Implementations compliant ~~to~~ with NVM Express Base Specification, R~~r~~evision 1.2 or later shall report a non-zero value in this field.</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td></tr>
<tr><td>331:328</td><td>O</td><td>O</td><td>R</td><td><b>Sanitize Capabilities (SANICAP):</b> This field indicates attributes for sanitize operations. If the Sanitize command is supported, then this field shall be non-zero. If the Sanitize command is not supported, then this field shall be cleared to 0h. Refer to section 8.21.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:30</td><td><b>No-Deallocate Modifies Media After Sanitize (NODMMAS):</b> This field indicates if media is additionally modified by the controller after a sanitize operation successfully completes that had been started by a Sanitize command with the No-Deallocate After Sanitize bit set to ‘1’.
...
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>Additional media modification after sanitize operation completes successfully is not defined. Only controllers compliant with NVM Express Base Specification, R~~r~~evision 1.3 and earlier or that have bits 2:0 of the SANICAP field cleared to 000b ~~shall be~~ are allowed to return this value.</td></tr>
<tr><td>...</td><td></td></tr>
</table>
If bits 2:0 of the SANICAP field are cleared to 000b, then the controller shall clear this field to 00b.</td></tr>
<tr><td>...</td><td></td></tr>
</table></td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5"><b>Fabric Specific</b></td></tr>
<tr><td>1795:1792</td><td>M[2]</td><td>M[2]</td><td>R</td><td><b>I/O Queue Command Capsule Supported Size (IOCCSZ):</b> This field defines the maximum I/O command capsule size in 16 byte units. The minimum value that shall be indicated is 4 corresponding to 64 bytes.</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td></tr>
<tr><td>2047:1806</td><td></td><td></td><td></td><td>Reserved</td></tr>
<tr><td colspan="5"><b>Power State Descriptors</b></td></tr>
<tr><td>2079:2048</td><td>M</td><td>M</td><td>R</td><td><b>Power State 0 Descriptor (PSD0):</b> This field indicates the characteristics of power state 0. The format of this field is defined in Figure 276.</td></tr>
<tr><td>...</td><td></td><td></td><td></td><td></td></tr>
<tr><td colspan="5">NOTES:
1. O/M/R definition: O = Optional, M = Mandatory, R = Reserved.
2. Mandatory for I/O controllers using a message-based transport. Reserved for ~~I/O~~ controllers using a memory-based transport.</td></tr>
</table>

...

Modify a portion of section 5.23 as shown below:

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

5.23 Namespace Management command

…

5.23.2 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 (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 301.

Figure 301: Namespace Management – Command Specific Status Values

Value	Description
0Ah	**Invalid Format:** The User Data Format specified is not supported. This may be due to various conditions, including: 1) specifying an invalid User Data Format number; 2) enabling protection information when there are not sufficient resources (e.g. 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)~~.
15h	**Namespace Insufficient Capacity:** Creating the namespace requires more unallocated capacity than is currently available. 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 8.12).~~
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 275)). 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 5.24 as shown below:

5.24 Sanitize command

…

If any Persistent Memory Region is enabled in an NVM subsystem, then the controller shall abort any Sanitize command with a status code of Sanitize Prohibited While Persistent Memory Region is Enabled. A sanitize operation is prohibited while ~~the~~ any Persistent Memory Region is enabled.

If any namespace is write protected in an NVM subsystem (refer to section 8.12), then the controller aborts any Sanitize command with a status code of Namespace is Write Protected. A sanitize operation is prohibited while any namespace is write protected.

If a firmware activation with reset is pending, then the controller shall abort any Sanitize command.
…

5.24.1 Command Completion

…

Sanitize command specific status values (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 305.

Figure 305: Sanitize – Command Specific Status Values

Value	Description
…	
11h	**Firmware Activation Requires Controller Level Reset:** The sanitize operation could not be started because a firmware activation is pending and a Controller Level Reset is required.
~~20h~~	~~**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to section 8.12)~~
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 5.27 as shown below:

5.27 Set Features command

…
<Editors note: The reference to 3.1.2 below should be made a hot link.>

5.27.1 Feature Specific Information

Figure 316 defines the Features that may be configured with a Set Features command and retrieved with a Get Features command. Refer to section 3.1.2 for mandatory, optional, and prohibited features for the various controller types.

5.27.1.18 Host Behavior Support (Feature Identifier 16h)

…

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

5.27.1.23 Host Metadata (Feature Identifier 7Dh), (Feature Identifier 7Eh), (Feature Identifier 7Fh)

…

Figure 361: Metadata Element Descriptor

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>04:00</td><td><b>Element Type (ET):</b> This field specifies the type of metadata stored in the descriptor.<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00h</td><td>Reserved</td></tr><tr><td>01h to ~~0~~17h</td><td>Element Types defined by this specification. Enhanced Controller Metadata Element and Controller Metadata Element types are defined in Figure 362. Namespace Metadata Element types are defined in Figure 363.</td></tr><tr><td>18h to 1Fh</td><td>Vendor Specific</td></tr></table></td></tr>
</table>

…

5.27.1.23.2 Controller Metadata (Feature Identifier 7Eh)

…

The Controller Metadata Feature provides backward compatibility with Management Controllers (refer to the NVM~~e~~ Express Management Interface Specification) compliant ~~to~~ with version 1.1 and earlier versions of the NVM~~e~~ Express Management Interface Specification.

…

Modify a portion of section 6.1 as shown below:

6.1 Authentication Receive Command and Response

…

Figure 377: Authentication Receive Response

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>07:00</td><td>Reserved</td></tr>
<tr><td>09:08</td><td><b>SQ Head Pointer (SQHD):</b> Indicates the current Submission Queue Head pointer for the associated Submission Queue.</td></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>13:12</td><td><b>Command Identifier (CID):</b> Indicates the identifier of the command that is being completed.</td></tr>
<tr><td>15:14</td><td><b>Status (STS):</b> ~~Specifies~~Indicates status for the command.<table><tr><th>Bits</th><th>Definition</th></tr><tr><td>15:01</td><td>Status field for the command. Refer to section 3.3.3.2.1.</td></tr><tr><td>00</td><td>Reserved</td></tr></table></td></tr>
</table>

…

Modify a portion of section 6.2 as shown below:

6.2 Authentication Send Command and Response

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 379: Authentication Send Response

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>07:00</td><td>Reserved</td></tr>
<tr><td>09:08</td><td><b>SQ Head Pointer (SQHD):</b> Indicates the current Submission Queue Head pointer for the associated Submission Queue.</td></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>13:12</td><td><b>Command Identifier (CID):</b> Indicates the identifier of the command that is being completed.</td></tr>
<tr><td>15:14</td><td><b>Status (STS):</b> ~~Specifies~~Indicates status for the command.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>15:01</td><td>Status field for the command. Refer to section 3.3.3.2.1.</td></tr>
<tr><td>00</td><td>Reserved</td></tr>
</table></td></tr>
</table>

…

Modify a portion of section 6.3 as shown below:

<Note to editor: The following section number reference (section 6) are plain text and need to be hot linked to that section.>

6.3 Connect Command and Response

…

If an NVM subsystem supports DH-HMAC-CHAP authentication (refer to section 6), then the Host NQN and the NVM Subsystem NQN parameters in a Connect command are required to be different. If the Host NQN and the NVM Subsystem NQN parameters in a Connect command are identical and the NVM subsystem supports DH-HMAC-CHAP authentication, then the controller shall abort the command with a status code of Connect Invalid Host ~~is returned~~.

The NVM subsystem shall not allocate a Controller ID in the range FFF0h to FFFFh as a valid Controller ID on completion of a Connect command. If the host is not allowed to establish an association to any controller in the NVM subsystem, then the controller shall abort the command with a status code of Connect Invalid Host ~~is returned~~.

If the NVM subsystem supports the dynamic controller model, then:

- the Controller ID of FFFFh ~~shall be~~ is specified as the Controller ID in a Connect command for the Admin Queue. If the controller ID is not set to FFFFh, then the controller shall abort the command with a status code of Connect Invalid Parameters ~~is returned~~;
- the NVM subsystem shall allocate any available controller to the host; and
- return that allocated Controller ID in the Connect response.

If the NVM subsystem supports the static controller model, then:

- The host ~~may~~ is able to request a specific controller in a Connect command for the Admin Queue. If the host is not allowed to establish an association to the specified controller, then the controller shall abort the command with a status code of Connect Invalid Host ~~is returned~~;
- The Controller ID of FFFEh on the Admin Queue specifies that any Controller ID may be allocated and returned in the Connect response; and
- If the host specifies a Controller ID value of FFFFh for the Admin Queue, then the controller shall abort the command with a status code of Connect Invalid Parameters ~~is returned~~.

The NVM subsystem may allocate specific controllers to particular hosts. If a host requests a controller that is not allocated to that host, then the controller shall abort the command with a status code of Connect

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Invalid Host ~~is returned~~. The mechanism for allocating specific controllers to particular hosts is outside the scope of this specification.

The host shall establish an association with a controller and enable the controller before establishing a connection with an I/O Queue of the controller. If the host sends a Connect command specifying a Queue ID for an Admin Queue or I/O Queue ~~which~~ that has already been created, then the controller shall abort the command with a status code of Command Sequence Error ~~is returned~~.

~~A status code of Connect Invalid Parameters is returned for~~ The controller shall abort a Connect command with a status code of Connect Invalid Parameters if:

- the host sends a Connect command to create an I/O Queue while the controller is disabled;
- …

…

If this situation occurs and ~~a status code is returned for~~ the Connect command is aborted, then ~~that~~ the status code ~~is~~ shall be set to Connect Invalid Parameters. There is no requirement that such a Connect command be received by an NVM subsystem (e.g., if the NVMe Transport Address is not a valid transport address, or is the address of a fabric endpoint that does not support NVMe over Fabrics, then the resulting error, if any, is specific to the fabric).

…

Figure 380: Connect Command – Submission Queue Entry

Bytes	Description
…	
43:42	**Queue ID (QID):** Specifies the Queue Identifier for the Admin Queue or I/O Queue to be created. The identifier is used for both the Submission and Completion Queue. The identifier for the Admin Submission Queue and Completion Queue is 0h. The identifier for an I/O Submission and Completion Queue is in the range 1 to 65,534. If the value in this field specifies the Queue ID of a queue that already exists, then the controller shall abort the command with a status code of Invalid Queue Identifier.
…	

…

Figure 382: Connect Response

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>15:14</td><td><b>Status (STS):</b> ~~Specifies~~Indicates status for the command. ~~Refer to Figure 97 for values specific to the Connect command.~~
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>15:01</td><td>Status field for the command. Refer to section 3.3.3.2.1. Refer to Figure 97 for values specific to the Connect command.</td></tr>
<tr><td>00</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

…

Modify a portion of section 6.4 as shown below:

6.4 Disconnect Command and Response

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 385: Disconnect Response

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>07:00</td><td>Reserved</td></tr>
<tr><td>09:08</td><td><b>SQ Head Pointer (SQHD):</b> Indicates the current Submission Queue Head pointer for the associated Submission Queue.</td></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>13:12</td><td><b>Command Identifier (CID):</b> Indicates the identifier of the command that is being completed.</td></tr>
<tr><td>15:14</td><td><b>Status (STS):</b> ~~Specifies~~Indicates status for the command. ~~Refer to Figure 97 for values specific to the Disconnect command.~~
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>15:01</td><td>Status field for the command. Refer to section 3.3.3.2.1. Refer to Figure 97 for values specific to the Disconnect command.</td></tr>
<tr><td>00</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

…

Modify a portion of section 6.5 as shown below:

6.5 Property Get Command and Response

…

Figure 387: Property Get Response

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>07:00</td><td><b>Value (VALUE):</b> ~~Specifes~~ Indicates the value returned for the property if the Property Get command is successful. If the size of the property is four bytes, then the value is specified in bytes 03:00 and bytes 07:04 are reserved.</td></tr>
<tr><td>09:08</td><td><b>SQ Head Pointer (SQHD):</b> Indicates the current Submission Queue Head pointer for the associated Submission Queue.</td></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>13:12</td><td><b>Command Identifier (CID):</b> Indicates the identifier of the command that is being completed.</td></tr>
<tr><td>15:14</td><td><b>Status (STS):</b> ~~Specifies~~Indicates status for the command.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>15:01</td><td>Status field for the command. Refer to section 3.3.3.2.1.</td></tr>
<tr><td>00</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

…

Modify a portion of section 6.6 as shown below:

6.6 Property Set Command and Response

…

Figure 389: Property Set Response

Bytes	Description
07:00	Reserved
09:08	**SQ Head Pointer (SQHD):** Indicates the current Submission Queue Head pointer for the associated Submission Queue.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 389: Property Set Response

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>11:10</td><td>Reserved</td></tr>
<tr><td>13:12</td><td><b>Command Identifier (CID):</b> Indicates the identifier of the command that is being completed.</td></tr>
<tr><td>15:14</td><td><b>Status (STS):</b> ~~Specifies~~Indicates status for the command.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>15:01</td><td>Status field for the command. Refer to section 3.3.3.2.1.</td></tr>
<tr><td>00</td><td>Reserved</td></tr>
</table></td></tr>
</table>

…

Modify a portion of section 7.2 as shown below:

7.2 Reservation Acquire command

…

Figure 392: Reservation Acquire – Command Dword 10

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>02:00</td><td><b>Reservation Acquire Action (RACQA):</b> This field specifies the action that is performed by the command.
<table>
<tr><th>RACQA Value</th><th>Description</th><th>Reference</th></tr>
<tr><td>000b</td><td>Acquire</td><td>8.19.5</td></tr>
<tr><td>001b</td><td>Preempt</td><td>8.19.7</td></tr>
<tr><td>010b</td><td>Preempt and Abort</td><td>8.19.7</td></tr>
<tr><td>011b to 111b</td><td colspan="2">Reserved</td></tr>
</table></td></tr>
</table>

…

Modify a portion of section 7.3 as shown below:

7.3 Reservation Register command

…

Figure 396: Reservation Register – Command Dword 10

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>02:00</td><td><b>Reservation Register Action (RREGA):</b> This field specifies the registration action that is performed by the command.
<table>
<tr><th>RREGA Value</th><th>Description</th><th>Reference</th></tr>
<tr><td>000b</td><td>Register Reservation Key</td><td>8.19.3</td></tr>
<tr><td>001b</td><td>Unregister Reservation Key</td><td>8.19.4</td></tr>
<tr><td>010b</td><td>Replace Reservation Key</td><td>8.19.3</td></tr>
<tr><td>011b to 111b</td><td colspan="2">Reserved</td></tr>
</table></td></tr>
</table>

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Modify a portion of section 7.4 as shown below:

7.4 Reservation Release command

…

Figure 399: Reservation Release – Command Dword 10

Bits	Description
…	
02:00	**Reservation Release Action (RRELA):** This field specifies the reservation action that is performed by the command. <table><tr><th>Value</th><th>Description</th><th>Reference</th></tr><tr><td>000b</td><td>Release</td><td>8.19.6</td></tr><tr><td>001b</td><td>Clear</td><td>8.19.8</td></tr><tr><td>010b to 111b</td><td>Reserved</td><td></td></tr></table>

…

Modify a portion of section 8.3 as shown below:

8.3 Capacity Management

8.3.1 Overview

…

The host uses Variable Capacity Management to:

- create a single Endurance Group by specifying the desired capacity;
- create a single NVM Set by specifying the desired capacity;
- delete a single Endurance Group; and
- delete a single NVM ~~set~~ Set.

…

8.3.3 Variable Capacity Management

…

A controller supporting Variable Capacity Management:

…

g) if NVM ~~sets~~ Sets are supported:

…

Modify a portion of section 8.7 as shown below:

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

8.7 Directives

Directives is a mechanism to enable host and NVM subsystem or controller information exchange. The Directive Receive command (refer to section 5.10) is used to transfer data related to a specific Directive Type from the controller to the host. The Directive Send command (refer to section 5.11) is used to transfer data related to a specific Directive Type from the host to the controller. Other commands may include a Directive Specific value specific for a given Directive Type (e.g., the Write command in the NVM ~~command set~~ Command Set).

…

8.7.1 Directive Use in I/O Commands

I/O Command Directives are the subset of Directive Types that may be used as part of I/O commands. For example, a Write command in the NVM ~~command set~~ Command Set may specify a Directive Type and an associated Directive Specific value. I/O Command Directives shall have a Directive Type value that is less than or equal to 0Fh due to the size of the Directive Type field in I/O commands. When a Directive Type is specified in an I/O command, the most significant four bits are assumed to be 0h. A Directive Type of 00h in an I/O command specifies that the I/O command is not using Directives.

…

8.7.3.1.1 Return Parameters (Directive Operation 01h)

The Return Parameter operation returns a data structure that specifies the features and capabilities supported by the Streams Directive, including namespace specific values. The DSPEC field in command Dword 11 is not used for this operation. The data structure returned is defined in Figure 425. If an NSID value of FFFFFFFFh is specified, then the controller:

- returns the NVM subsystem specific values~~,~~;
- may return any namespace specific values that are the same for all namespaces (e.g., SWS)~~,~~; and
- clears all other namespace specific fields to 0h.

…

Modify a portion of section 8.22 as shown below:

8.22 Submission Queue (SQ) Associations

…

The host enables the SQ Associations capability by creating an association between an NVM Set and a Submission Queue at the time the Submission Queue is created (e.g., with a Create I/O Submission Queue command (refer to section 5.5)).

~~In order for~~ For the SQ Associations capability to yield benefits, the host is required to:

…

Modify a portion of section 8.24 as shown below:

8.24 Telemetry

...

There are two telemetry data logs (i.e., Telemetry Host-Initiated log page and Telemetry Controller-Initiated log page) defined. Each telemetry data log is made up of a single set of Telemetry Data Blocks. Each Telemetry Data Block is 512 bytes in size.

…

The preparation, collection, and submission of telemetry data is similar for host-initiated and controller-initiated data; the primary difference is the trigger for the collection. The operational model for telemetry is:

1. The host identifies controller support for Telemetry log pages in the Identify Controller data structure;

…

5. If the host decides to collect host-initiated telemetry data or the controller signals that controller-initiated telemetry data is available:
    a. The host reads the appropriate blocks of the Telemetry Data Area from the Telemetry ~~h~~Host-~~i~~Initiated log page (refer to section 5.16.1.8) or the Telemetry ~~c~~Controller-~~i~~Initiated log page (refer to section 5.16.1.9). If possible, the host should collect Telemetry Data Area 1, 2, 3, and 4. The host reads the log in 512 byte Telemetry Data Block units (i.e., a starting offset that is a multiple of 512, and a length that is a multiple of 512). The host should set the Retain Asynchronous Event bit to ‘1’;
    b. The host re-reads the header of the log page and ensures that the Telemetry Host-Initiated Data Generation Number field from the Telemetry ~~h~~Host-~~i~~Initiated log page or the Telemetry Controller-Initiated Data Generation Number field in the Telemetry ~~c~~Controller-~~i~~Initiated log page matches the original value read. If these values do not match, then the data captured is not consistent and should be re-read from the log page with the Retain Asynchronous Event bit set to ‘1’;
    c. If the host is reading the Telemetry ~~c~~Controller-~~i~~Initiated log page, then the host reads any portion of that log page with the Retain Asynchronous Event bit cleared to ‘0’ to indicate to the controller that the host has completed reading the Telemetry ~~c~~Controller-~~i~~Initiated log page; and

    …

…

The NVM subsystem is allowed to provide a Telemetry ~~h~~Host-~~i~~Initiated log page per controller or a shared Telemetry ~~h~~Host-~~i~~Initiated log page across all controllers in the NVM subsystem. ~~When~~ If a shared Telemetry ~~h~~Host-~~i~~Initiated log page is implemented, the Telemetry Host-Initiated Data Generation Number field in the Telemetry ~~h~~Host-~~i~~Initiated log page is used to allow the host to detect that the Telemetry Host-Initiated log page has been changed by a host ~~from~~ through a different controller.

…

Modify a portion of section B.4 as shown below:

<Editors note: The reference to step 12 below should be made a hot link so a future change to that list doesn’t make this reference wrong.>

B.4 Asynchronous Event Request Host Software Recommendations

This section describes the recommended host software procedure for Asynchronous Event Requests.

The host sends *n* Asynchronous Event Request commands (refer to section 3.5.1, step ~~10~~12). When an Asynchronous Event Request completes (providing Event Type, Event Information, and Log Page details):
…

Description of NVM Express® NVM Command Set Specification changes

Modify a portion of section 1.6 as shown below:

<Note to editor: Change “http” to “https” in the first web link – change both displayed text and hyper link.>

1.6 References

NVM Express Base Specification, Revision 2.0. Available from https://www.nvmexpress.org.

NVM Express Management Interface Specification, Revision 1.2. Available from https://www.nvmexpress.org.
…

Modify a portion of section 2.1 as shown below:

2.1.1 Namespaces

…

~~When~~ If the THINP bit in the NSFEAT field of the Identify Namespace data structure is set to ‘1’, the controller:
…

~~When~~ If the THINP bit is cleared to ‘0’, the controller:
…

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

- a Write command (refer to section 3.2.6);
- a Write Uncorrectable command (refer to section 3.2.7); or
- a Write Zeroes command (refer to section 3.2.8) that does not deallocate the logical block ~~(refer to section 3.2.3.2.1)~~.

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 3.2.3.2.1)~~.

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

- a Dataset Management command (refer to section 3.2.3); ~~or~~
- a Write Zeroes command addressing the logical block ~~(refer to section 3.2.3.2.1)~~; or
- a sanitize operation.

…

Modify a portion of section 3.1 as shown below:

3.1.2 NVM Command Set Specific Status Values

Figure 15: Status Code – Generic Command Status Values

Value	Description
14h	**Atomic Write Unit Exceeded:** The length specified exceeds the atomic write unit size.
1Eh	**SGL Data Block Granularity Invalid:** The Address alignment or Length granularity for an SGL Data Block descriptor is invalid. This may occur when a controller supports dword granularity only and the least significant two bits of the Address or Length are not cleared to 00b. Note: An implementation compliant ~~to~~ with revision 1.2.1 of the NVMe Base Specification or earlier may use the status code value of 15h to indicate SGL Data Block Granularity Invalid.
80h	**LBA Out of Range:** The command references an LBA that exceeds the size of the namespace.

…

Modify a portion of section 3.2 as shown below:

3.2 NVM Command Set Commands

…

3.2.3 Dataset Management command

…

3.2.3.3 Command Completion

…

Dataset Management command specific status values (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 43.

Figure 43: 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 the Namespace Write Protection section of the NVMe Base Specification).~~
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 the Namespace Write Protection section in the NVMe Base Specification).
83h	**Command Size Limit Exceeded:** One or more of the Dataset Management processing limits (i.e., non-zero values of the DMRL, DMRSL and DMSL fields in the Identify Controller data structure) was exceeded (refer to section 3.2.3.1). The controller shall not return this status value if bit 2 is set to '1' in the Optional NVM Command Support field in the Identify Controller data structure.

…

3.2.6 Write command

…

3.2.6.1 Command Completion

…

Write command specific errors (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 67.

Figure 67: Write – Command Specific Status Values

Value	Description
~~20h~~	~~**Namespace is Write Protected:** The command is prohibited while the namespace is write protected (refer to the Namespace Write Protection section of the NVMe Base Specification).~~
80h	**Conflicting Attributes:** The attributes specified in the command are conflicting.
81h	**Invalid Protection Information:** The Protection Information Field (PRINFO) (refer to Figure 63) settings specified in the command are invalid for the Protection Information with which the namespace was formatted (refer to the PI field in Figure 78 and the DPS field in Figure 97) or the ILBRT field is invalid (refer to section 5.2.2.5).
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 the Namespace Write Protection section in the NVMe Base Specification).

…

3.2.7 Write Uncorrectable command

…

3.2.7.1 Command Completion

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

Write Uncorrectable command specific errors (i.e., SCT field set to 1h) are shown in Figure 70.

Figure 70: 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 the Namespace Write Protection section of the NVMe Base Specification).~~
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 the Namespace Write Protection section in the NVMe Base Specification).

<Editors note: In the following section, the reference to 5.2.1.4.1 should be made “hot”.>

3.2.8 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 97), CDW15.LBATM, CDW15.LBAT, as well as CDW2/3 and CDW14 content as described in section 5.2.1.4.1. Protection information of all zeroes is generated if the PRACT bit is cleared to 0h resulting in invalid protection information; therefore, the host should set the PRACT bit to ‘1’ to generate valid protection information.

After successful completion of this command, the value returned by subsequent successful reads of logical blocks and associated metadata (excluding protection information) in this range shall be all bytes cleared to 0h until a write occurs to this LBA range.

~~If the Deallocate bit (CDW12.DEAC) is set to '1' in a Write Zeroes command, and the namespace supports clearing all bytes to 0h in the values read (e.g., bits 2:0 in the DLFEAT field are set to 001b) from a deallocated logical block and its metadata (excluding protection information), then for each specified logical block, the controller:~~

- ~~should deallocate that logical block;~~
- ~~shall return all bytes cleared to 0h in the values read from:~~
  - ~~that logical block; and~~
  - ~~that logical blocks metadata (excluding protection information);~~

  ~~and~~
- ~~shall return the protection information in that logical block as specified in section 3.2.3.2.1.~~

~~If the Deallocate bit is cleared to '0' in a Write Zeroes command, and the namespace supports clearing all bytes to 0h in the values read (e.g., bits 2:0 in the DLFEAT field are set to 001b) from a deallocated logical block and its metadata (excluding protection information), then, for each specified logical block, the controller:~~

- ~~may deallocate that logical block;~~
- ~~shall return all bytes cleared to 0h in the values read from:~~
  - ~~that logical block; and~~
  - ~~that logical blocks metadata (excluding protection information);~~

  ~~and~~
- ~~shall return the protection information in that logical block based on CDW12.PRINFO in that Write Zeroes command.~~

For each logical block in the range specified by a Write Zeroes command, if the namespace supports clearing all bytes to 0h in the values read (e.g., bits 2:0 in the DLFEAT field are set to 001b) from a deallocated logical block and its metadata (excluding protection information), and the value of the Deallocate bit (CDW12.DEAC) in that Write Zeroes command is:

- set to '1', then the controller should deallocate that logical block; and
- cleared to '0', then the controller may deallocate that logical block.

For each logical block in the range specified by a Write Zeroes command, if the namespace does not support ~~that logical block~~ clearing all bytes to 0h in the values read from that logical block and its metadata (excluding the protection information) when that logical block is deallocated ~~read~~, then the controller shall not deallocate that logical block.

If a logical block in the range specified by a Write Zeroes command is deallocated as a result of that command, then:

- the DULBE bit in the Error Recovery feature (refer to section 4.1.3.2) affects whether subsequent reads of that deallocated logical block are able to succeed (refer to section 3.2.3.2.1); and
- the values of protection information, if any, returned by subsequent successful reads of that deallocated logical block are specified in section 3.2.3.2.1.

If a logical block in the range specified by a Write Zeroes command is not deallocated as a result of that command, then the values of protection information, if any, returned by subsequent successful reads of that logical block shall be based on CDW12.PRINFO in that Write Zeroes command.

If the Write Zeroes Size Limit (WZSL) field in the Identify Controller data structure is set to a non-zero value,

…

3.2.8.1 Command Completion

…

Write Zeroes command specific status values (i.e., SCT field set to 1h) are ~~defined~~ shown in Figure 76.

Figure 76: 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 the Namespace Write Protection section of the NVMe Base Specification).~~
81h	**Invalid Protection Information:** The Protection Information Field (PRINFO) (refer to Figure 73) settings specified in the command are invalid for the Protection Information with which the namespace was formatted (refer to the PI field in Figure 78 and the DPS field in Figure 97) or the ILBRT field is invalid (refer to section 5.2.2.5).
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 the Namespace Write Protection section in the NVMe Base Specification).

…

Modify a portion of section 4.1 as shown below:

4.1 Admin Command behavior for the NVM Command Set

…

4.1.5 Identify Command

…

<Editors note: Correct the table heading line to make it the first row of the table – so it propagates correctly onto subsequent pages. Also correct footnote reference font size to 14 point. Also correct reference hot links – they appear hot, but don't go anywhere.>

Figure 96: CNS Values

CNS Value	O/M [1]	Definition	NSID [2]	CNTID [3]	CSI [4]	Reference Section
Active Namespace Management						
00h	M	Identify I/O Command Set Specific Namespace data structure for the controller processing the command. [6]	Y	N	~~Y~~N	4.1.5.1
…	M	Identify Controller data structure for the controller processing the command. [6]	N	N	N	4.1.5.2
05h	M [5]	Identify I/O Command Set specific Namespace data structure for the specified NSID for the I/O Command Set specified in the CSI field. [6]	Y	N	Y	4.1.5.3
06h	M	Identify I/O Command Set s~~S~~pecific Controller data structure for the controller processing the command. [6]	Y	N	Y	4.1.5.4
11h	O	Identify Namespace data structure for the specified allocated NSID.	Y	N	N	4.1.5.5
16h	O	A Namespace Granularity List (refer to Figure ~~X~~ 103) is returned to the host ~~for~~ that contains up to sixteen Namespace Granularity Entries.	N	N	N	4.1.5.6

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

CNS Value	O/M [1]	Definition	NSID [2]	CNTID [3]	CSI [4]	Reference Section
NOTES: 1. O/M definition: O = Optional, M = Mandatory. 2. The NSID field is used: Y = Yes, N = No. 3. The CDW10.CNTID field is used: Y = Yes, N = No. 4. The CDW11.CSI field is used: Y = Yes, N = No. 5. Mandatory for controllers that support the Namespace Management capability (refer to the NVMe Base Specification). 6. Selection of a UUID may be supported. Refer to the Universally Unique Identifiers (UUIDs) for Vendor Specific Information section in the NVMe Base Specification.						

…

4.1.5.1 NVM Command Set Identify Namespace Data Structure (CNS 00h)

…

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

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td>…</td><td></td><td></td></tr>
<tr><td>28</td><td>M</td><td><b>End-to-end Data Protection Capabilities (DPC):</b> This field indicates the capabilities for the end-to-end data protection feature. Multiple bits may be set in this field. Refer to section <b>Error! Reference source not found.</b>.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>…</td><td></td></tr>
<tr><td>3</td><td><b>Protection Information In First Bytes (PIIFB):</b> If set to ‘1’ indicates that the namespace supports protection information transferred as the first bytes of metadata. If cleared to ‘0’ indicates that the namespace does not support protection information transferred as the first bytes of metadata. For implementations compliant ~~to~~ with revision 1.0 or later of the NVM Command Set Specification, this bit shall be cleared to ‘0’.</td></tr>
<tr><td>…</td><td></td></tr>
</table></td></tr>
<tr><td>29</td><td>M</td><td><b>End-to-end Data Protection Type Settings (DPS):</b> This field indicates the protection information Type settings for the end-to-end data protection feature. Refer to section 5.2.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7:4</td><td>Reserved</td></tr>
<tr><td>3</td><td><b>Protection Information Position (PIP):</b> This bit, if set to ‘1’, indicates that the protection information, if enabled, is transferred as the first bytes of metadata. This bit ~~Bit 3~~, if cleared to ‘0’, indicates that the protection information, if enabled, is transferred as the last bytes of metadata. For implementations compliant ~~to~~ with version 1.0 or later of the NVM Command Set Specification, this bit shall be cleared to ‘0’.</td></tr>
<tr><td>…</td><td></td></tr>
</table></td></tr>
<tr><td>…</td><td></td><td></td></tr>
</table>

…

4.1.6 Namespace Management command

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 105: Namespace Management – Host Software Specified Fields

Bytes	Description	Host Specified
Fields that are a subset of the Identify Namespace data structure (refer to Figure 97)		
…		
95:92	ANA Group Identifier (ANAGRPID) [1]	Yes
99:96	Reserved	
101:100	NVM Set Identifier (NVMSETID) [1]	Yes
103:102	Endurance Group Identifier (ENDGID) [1]	Yes
383:104	Reserved	
…		

Notes:

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

…

Modify a portion of section 5.2 as shown below:

5.2 End-to-end Data Protection

To provide robust data protection from the application to the NVM media and back to the application itself, end-to-end data protection may be used. If this optional mechanism is enabled, then additional protection information (e.g., CRC) is added to the logical block that may be evaluated by the controller and/or host software to determine the integrity of the logical block. This additional protection information, if present, is either the first bytes of metadata or the last bytes of metadata, based on the format of the namespace (refer to the PIP bit in the DPS field shown in Figure 97). If the Metadata Size (refer to Figure 101) is greater than the number of bytes of protection information and the protection information is contained in the first bytes of the metadata, then the CRC does not cover any metadata bytes. If the Metadata Size is greater than the number of bytes of protection information and the protection information is contained in the last bytes of the metadata, then the CRC covers all metadata bytes up to but excluding the protection information. As described in section 5.8.3, metadata and hence this protection information may be configured to be contiguous with the logical block data or stored in a separate buffer.

…

5.2.1.2 32b Guard Protection Information

…

The Guard field contains a 32b CRC computed over the logical block data. The formula used to calculate the CRC is the CRC-32C (Castagnoli) which uses the generator polynomial 1EDC6F41h (refer to the NVM Express Management Interface ~~s~~Specification). The Application Tag and Storage and Reference Space fields have the same definition as defined by 16b Guard Protection Information (refer to section 5.2.1.1).

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

5.2.1.3 64b Guard Protection Information

The 64b Guard Protection Information is shown in Figure 118 and is contained in the metadata associated with each logical block. 64b Guard Protection Information shall only be available to namespaces that have an LBA size (refer to the LBADS field in Figure 98) greater than or equal to 4 KiB.

The Guard field contains a 64b CRC computed over the logical block data. The polynomial used to calculate the CRC is defined in section 5.2.1.3.1 ~~Figure 118~~. The Application Tag and Storage and Reference Space have the same definition as defined by 16b Guard Protection Information (refer to section 5.2.1.1).

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Description of NVM Express® Zoned Namespace Command Set Specification changes

Modify a portion of section 2.1 as shown below:

2.1 Theory of operation

…

2.1.1.1 Zone Descriptor

…

Figure 3: Summary of Zone Descriptor Attributes

Attribute	Description
…	
Write Pointer	The Write Pointer attribute defines the ~~next~~ lowest numbered writeable logical block address in that zone. The validity of the write pointer is zone state specific and is defined per zone type (refer to section 2.1.1.2).
…	

…

2.1.1.4.1 Managing resources

…

A controller processing a command that requests a zone to transition to the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state shall, if resources are not available and:

a) the Maximum Active Resources field is greater than the Maximum Open Resources field and:

   i. as a result of the requested transition the resource that is not available is Active Resources, then abort the command with a status code of Too Many Active Zones; or

   ii. as a result of the requested transition the resource that is not available is Open Resources, and:

      a. one or more zones are in the ZSIO:Implicitly Opened state, then the controller shall select one of the zones in the ZSIO:Implicitly Opened state, transition that zone to the ZSC:Closed state, and proceed to process the command; or

      b. no zones are in the ZSIO:Implicitly Opened state, then the controller shall abort the command with a status code of Too Many Open Zones;

   or

b) the Maximum Active Resources field is equal to the Maximum Open Resources field and as a result of the requested transition the resource that is not available is Active Resources, then abort the command with a status code of Too Many Active Zones.

Zones that have associated Active Resources are transitioned to the ZSF:Full state when the zoned namespace becomes write protected. Refer to the Namespace Write Protection section in the NVMe Base Specification.

~~The controller may transition zones in the ZSIO:Implicitly Opened state to the ZSC:Closed state for resource management purposes.~~

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

The coordination of host software usage of resources associated with shared zoned namespaces is outside the scope of this specification.

…

Modify a portion of section 4.1 as shown below:

4.1.5 Identify Command

…

Figure 47: CNS Values

CNS Value	O/M [1]	Definition	NSID [2]	CNTID [3]	CSI [4]	Reference Section
Active Namespace Management						
00h	M	Identify Namespace data structure for the specified NSID or the common namespace capabilities for the NVM Command Set. [5]	Y	N	N [6]	NVM Command Set Specification
…						
05h	M	I/O Command Set specific Identify Namespace data structure for the specified NSID for the I/O Command Set specified in the CSI field. [5]	Y	N	Y	CSI 00h: NVM Command Set Specification CSI 02h: 4.1.5.1
06h	M	I/O Command Set specific Identify Controller data structure for the controller processing the command. [5]	Y	N	Y	4.1.5.2
…						

NOTES:

…

6. This Identify data structure applies to namespaces that are associated with command sets that specify logical blocks (i.e., Command Set Identifier 0h or Command Set Identifier 02h).

…

4.1.5.1 I/O Command Set Specific Identify Namespace Data Structure for the Zoned Namespace Command Set (CNS 05h, CSI 02h)

…

Figure 48: I/O Command Set Specific Identify Namespace Data Structure for the Zoned Namespace Command Set

Bytes	O/M [1]	Description
…		
07:04	M	**Maximum Active Resources (MAR):** This field defines the maximum number of ~~concurrently~~ active zones in the zoned namespace. A value of FFFFFFFFh indicates that there is no limit. This is a 0's based value
11:08	M	**Maximum Open Resources (MOR):** This field defines the maximum number of ~~concurrently~~ open zones in the zoned namespace. This field shall be less than or equal to the Maximum Active Resources field. A value of FFFFFFFFh indicates that there is no limit. This is a 0's based value
…		

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

4.1.7 Sanitize command

…

Figure 51: Sanitize Behavior for the Zoned Namespace Command Set

No Deallocate After Sanitize	No-Deallocate Modifies Media After Sanitize (NODMMAS)	No-Deallocate Inhibited (NDI)	No-Deallocate Response Mode (NODRM)	Results of a successful sanitize operation		
				Zone State [1]	Logical Block Content [2]	Sanitize Status [4]
…						
1b	10b (does modify)	0b	N/A [3]	ZSF:Full	Block Erase: not defined in this specification Crypto Erase: not defined in this specification ~~Overwrite:~~ Overwrite: Shall be the overwrite pattern specified in the Sanitize command that requested the sanitize operation	001b
…						

…

Modify a portion of sections 5.4 as shown below:

5.4 Reset Zone Recommended

…

If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:

a) clearing the Reset Zone Recommended zone attribute of the specific zone to ‘0’; and
b) generating a Zone Descriptor Changed event for the specific zone.

If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Reset Zone Recommended attribute shall be cleared to ‘0’.

Modify a portion of sections 5.5 as shown below:

5.5 Finish Zone Recommended

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

If the controller has processed the internal operation or the internal operation is no longer scheduled, the controller may notify the host by:

a) clearing the Finish Zone Recommended zone attribute of the specific zone to ‘0’; and
b) generating a Zone Descriptor Changed event for the specific zone.

If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Finish Zone Recommended zone attribute shall be cleared to ‘0’.

Modify a portion of sections 5.6 as shown below:

5.6 Zone Active Excursions

…

If the controller performs a Zone Active Excursion on such a zone, then the controller shall notify the host by:

a) setting the ~~zone attribute~~ Zone Finished by Controller ~~bit~~ zone attribute of that zone to ‘1’ (refer to Figure 37); and
b) generating a Zone Descriptor Changed event for that zone.

If a zone is in the ZSF:Read Only state or the ZSF:Offline state, then the Zone Finished by Controller attribute shall be cleared to ‘0’.

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Description of NVM Express® RDMA Transport Specification changes

Modify a portion of section 1.4 as shown below:

1.4.4 Inbound RDMA Read Queue Depth (IRD)

The maximum number of incoming outstanding RDMA_READ Requests that the RDMA-Capable Controller can handle on a particular RDMA-Capable Protocol Stream at the Data Source. Refer to ~~www.infinibandta.org~~ InfiniBand™ Architecture Specification Volume 1.

…

1.4.12 RDMA NIC (RNIC)

RDMA enabled network interface card (i.e., network adapter).

1.4.13 RDMA Private Data

Data that is opaque to the communication management protocol, passed from the sender to the recipient. Refer to ~~www.infinibandta.org~~ the term "private data" in InfiniBand™ Architecture Specification Volume 1.

…

1.4.18 RoCE and RoCEv2 RDMA

RDMA over Converged Ethernet definition. Refer to Annex A16 (RoCE) and Annex A17 (RoCE-v2) at www.infinibandta.org.

1.4.19 RNR_RETRY_COUNT

The total number of times that the "Request for Communication (REQ)" or the "Reply to Request for Communication (REP)" sender wishes the receiver to retry Receiver Not Ready (RNR) NAK errors before posting a completion error. Refer to ~~www.infinibandta.org~~ the term "RNR retry count" in InfiniBand™ Architecture Specification Volume 1.

Modify a portion of section 1.5 as shown below:

<Note to editor: Change "http" to "https" in the first web link – change both displayed text and hyper link.>

1.5 References

NVM Express® Base Specification, Revision 2.0. Available from https://www.nvmexpress.org.

InfiniBand™ Architecture Specification Volume 1. Available from https://www.infinibandta.org/ibta-specification/.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

RFC 5040, R. Recio, B. Metzler, P. Culley, J. Hilland, D. Garcia, “A Remote Direct Memory Access Protocol Specification”, October 2007. Available from https://tools.ietf.org/html/rfc5040.

RFC 5041, Shah, H., Pinkerton, J., Recio, R., and P. Culley, "Direct Data Placement over Reliable Transports", October 2007. Available from https://tools.ietf.org/html/rfc5041.

RFC5044, Culley, P., Elzur, U., Recio, R., Bailey, S., and J. Carrier, "Marker PDU Aligned Framing for TCP Specification", October 2007. Available from https://tools.ietf.org/html/rfc5044.

RFC6581, Kanevsky, A., Ed., Bestler, C., Ed., Sharp, R., and S. Wise, "Enhanced Remote Direct Memory Access (RDMA) Connection Establishment", April 2012. Available from https://tools.ietf.org/html/rfc6581.

Modify a portion of section 2 as shown below:

2 Transport Overview

…

In some host and NVM subsystem implementations, the interface between the RDMA transport and the RDMA providers is defined by an implementation of RDMA Verbs. When applicable, the host and NVM subsystem RDMA transport implementations should use the common RDMA Verbs software interfaces for the RDMA transport layer to be RDMA provider agnostic.

The iWARP RDMA transport over TCP/IP consists of the RDMAP, DDP and MPA protocols defined by the IETF (refer to IETF RFC 5040, RFC 5041, RFC 5044 and RFC 6581).

…

Modify a portion of section 3.1 as shown below:

3.1 Setup & Initialization

…

3.1.4 Fabric Dependent Settings

…

<editors note: Insert blank lines as shown in the CRQSIZE field below.>

Figure 6: RDMA_CM_ACCEPT Private Data Format

Bytes	Description
01:00	**Record Format (RECFMT):** Specifies the format of the RDMA Private Data. If a new format is defined, this value is incremented by one. The format of the record specified in this definition shall be 0h.

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 6: RDMA_CM_ACCEPT Private Data Format

Bytes	Description
03:02	**RDMA QP Controller Receive Queue Size (CRQSIZE):** This field indicates the number of RDMA QP receive queue entries allocated by the controller's RDMA Transport for capsule reception as a 1's based value. RDMA Transports that use RNR_RETRY_COUNT flow control may set this ~~entry~~ field to be less than or equal to the value ~~of~~ represented by the 0's based value in the HSQSIZE field specified in Figure 5 (e.g., if HSQSIZE is set to 3, then this field is set to a value less than or equal to 4). RDMA Transports that do not use RNR_RETRY_COUNT shall set this value to be equal to the value ~~of~~ represented by the 0's based value in the HSQSIZE field specified in Figure 5 (e.g., if HSQSIZE is set to 4, then this field is set to 5).
31:04	Reserved

Description of NVM Express® TCP Transport Specification changes

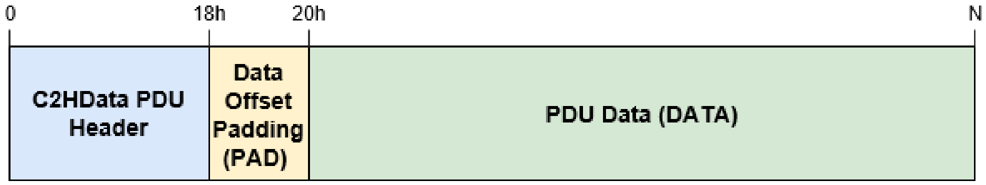
Modify a portion of section 3.3 as shown below:

3.3 Data Transfer Model

…

Figure 9 shows an example of a C2HData PDU where the HPDA field (refer to section 3.6.2.2) was set to 03h by the host in the ICReq when the connection was established. The C2HData PDU Header size 24 bytes and header digest is disabled. An alignment of 16 bytes is required, thus the controller inserts 8 bytes of padding in the PAD field.

Figure 9: Example of ~~64~~16B PDU DATA Alignment in C2HData PDU



The TCP transport defines NVMe/TCP PDU types that are summarized **…**

…

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Description of NVM Express® Management Interface Specification changes

Modify Title Page as shown below:

NVM Express®

Management Interface Specification

Modify a portion of section 3.1 as shown below:

3.1 NVMe-MI Messages

…

3.1.1 Message Fields

…

Figure 19: NVMe-MI Message Fields

Bytes	Description
…	

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 19: NVMe-MI Message Fields

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>2</td><td>
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7:2</td><td>Reserved</td></tr>
<tr><td>1</td><td>

Command Initiated Auto Pause (CIAP): If this bit is set to ‘1’ in a Command Message, the Management Endpoint shall be automatically paused when the Command Message enters the Process state. If this bit is cleared to ‘0’ in a Command Message, the Management Endpoint shall not be automatically paused when the Command Message enters the Process state. The usage requirements for this bit are as follows:

<table>
<tr><th colspan="2">Mechanism</th><th>CIAP Value</th><th>Usage Requirement</th></tr>
<tr><td rowspan="4">Out-of-band</td><td rowspan="2">Command Messages</td><td>0</td><td>This value is permitted.</td></tr>
<tr><td>1</td><td>This value is permitted.</td></tr>
<tr><td rowspan="2">Any NVMe-MI Message other than a Command Message</td><td>0</td><td>This value is required.</td></tr>
<tr><td>1</td><td>This value is prohibited. An Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.</td></tr>
<tr><td colspan="2" rowspan="2">In-band Tunneling</td><td>0</td><td>This value is required.</td></tr>
<tr><td>1</td><td>This value is prohibited. An Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.</td></tr>
</table>

~~This bit is only valid for Command Messages sent using the out-of-band mechanisms and is reserved for all other types of NVMe-MI Messages. If this bit is set to ‘1’ for any type of NVMe-MI Message received by a Management Endpoint other than a Command Message, then an Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.~~

~~If this bit is set to ‘1’ in the in-band tunneling mechanism, then an Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.~~

</td></tr>
<tr><td>0</td><td>

Management Endpoint Buffer (MEB): This bit indicates whether the Message Data in a Command Message is contained in the Message Data field of this NVMe-MI Message or in the Management Endpoint Buffer. Refer to section 3.1.

<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0b</td><td>The Message Data is contained in the Message Data of this NVMe-MI Message.</td></tr>
<tr><td>1b</td><td>The Message Data is contained in the Management Endpoint Buffer.</td></tr>
</table>

The usage requirements for this bit are as follows:

<table>
<tr><th colspan="2">Mechanism</th><th>MEB Value</th><th>Usage Requirement</th></tr>
<tr><td rowspan="4">Out-of-band</td><td rowspan="2">Command Messages</td><td>0</td><td>This value is permitted.</td></tr>
<tr><td>1</td><td>This value is permitted.</td></tr>
<tr><td rowspan="2">Any NVMe-MI Message other than a Command Message</td><td>0</td><td>This value is required.</td></tr>
<tr><td>1</td><td>This value is prohibited. An Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.</td></tr>
<tr><td colspan="2" rowspan="2">In-band Tunneling</td><td>0</td><td>This value is required.</td></tr>
<tr><td>1</td><td>This value is prohibited. An Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.</td></tr>
</table>

~~This bit is only valid for Command Messages sent using the out-of-band mechanism and is reserved for all other types of NVMe-MI Messages. If this bit is set to ‘1’ for any type of NVMe-MI Message received by a Management Endpoint other than a Command Message, then an Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.~~

~~If this bit is set to ‘1’ in the in-band tunneling mechanism, then an Invalid Parameter Error Response with the PEL field indicating this bit shall be returned.~~

</td></tr>
</table>
</td></tr>
</table>

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 19: NVMe-MI Message Fields

Bytes	Description
…	

…

Appendix A Technical Note: NVM Express Basic Management Command

…

Figure 175: Subsystem Management Data Structure

<table>
<tr><th>Command Code</th><th>Offset (byte)</th><th>Description</th></tr>
<tr><td rowspan="3">0</td><td>…</td><td></td></tr>
<tr><td>06</td><td>
<p><b>Current Power (Optional):</b> This field reports the current NVM Subsystem power consumption. If both bit mapped fields are cleared to 0h, then this field is not reported.</p>
<table>
<tr><th>Bit</th><th>Definition</th></tr>
<tr><td>7</td><td><b>NVM Subsystem Idle (NVMSI):</b> This bit is set to ‘1’ when the NVM Subsystem is idle and has been idle for at least 5 s. Refer to the NVMe Idle Power (IDLP) definition.</td></tr>
<tr><td>6:0</td><td>
<p><b>NVM Subsystem Power (NVMSP):</b> This field reports the ceiling function of the power consumed by the NVM Subsystem in w~~W~~atts. If ~~this value~~ the power consumed by the NVM Subsystem in watts is greater than or equal to 127 W, then 127 W is reported in this field.</p>
<p>Power reported by the NVM Subsystem is determined in the following manner. If the NVMSI bit is set to ‘1’, then the value returned in this field is:</p>
<ul>
<li>equal to ~~that~~ the value reported in the Idle Power (IDLP) field in the Power State Descriptor Data Structure for the corresponding NVMe power state if the IDLP field is set to a non-zero value; or</li>
<li>equal to the value reported in the Maximum Power (MP) field in the Power State Descriptor data structure for the corresponding NVMe power state, if the IDLP field is cleared to 0h.</li>
</ul>
<p>If the NVMSI bit is cleared to ‘0’, then the value returned in this field is:</p>
<ul>
<li>equal to ~~that~~ the value reported in the Active Power Workload (APW) field in the Power State Descriptor Structure for the corresponding NVMe power state if the APW field is set to a non-zero value; or</li>
<li>equal to the value reported in the Maximum Power (MP) field in the Power State Descriptor data structure for the corresponding NVMe power state, if the APW field is cleared to 0h. ~~The Maximum Power (MP) field value is substituted for IDLP or APW if these are not for reported in the Power State Descriptor Structure for the current NVMe power state.~~</li>
</ul>
</td></tr>
</table>
</td></tr>
<tr><td>…</td><td></td></tr>
<tr><td>…</td><td></td><td></td></tr>
</table>

Technical input submitted to the NVM Express® Workgroup is subject to the terms of the NVM Express® Participant’s agreement. Copyright © 2008 to 2022 NVM Express, Inc.