



LEGAL NOTICE:

© **Copyright 2007 - 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: “© 2007 - 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

NVM Express™ Technical Errata

Errata ID	102
Revision Date	1/6/2022
Affected Spec Ver.	NVM Express™ Base Specification Revision 2.0a NVM Express™ NVM Command Set Specification Revision 1.0a NVM Express™ Zoned Namespace Command Set Specification Revision 1.1a NVM Express™ Key Value Command Set Specification Revision 1.0a NVM Express™ PCI Express® Transport Specification Revision 1.0 NVM Express™ TCP Transport Revision 1.0
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Fred Knight	NetApp
Judy Brock, Mike Allison, Bill Martin	Samsung
Paul Suhler, John Geldman	Kioxia
Matias Bjorling, Yoni Shternhell	Western Digital
David Black, Austin Bolen	Dell/EMC
Curtis Ballard	HPE
Jim Hatfield	Seagate
Andres Baez	Intel

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 Key Value Command Set specification, the NVM Express PCI Express Transport specification, and the NVM Express TCP Transport specification. This ECN also builds on the content included in ECN104.

Revision History

Revision Date	Change Description
5/24/2021	Initial creation
6/8/2021	Remove KATO clarifications into TP4129, Integrate changes from email list; changes from ZNS 30 day review.
7/20/2021	Add Get LBA status corrections from Judy, bring in cache language that was pushed out from ECN-001.
7/22/2021	Updates from TWG
9/24/2021	Updates from TWG – move agreed content to ECN105.
10/6/2021	Update Description of changes sections and list of contributors
10/7/2021	Fix a reference ERROR!
11/2/2021	Incorporate 30-day review comments from Samsung.
11/4/2021	Incorporate feedback from WG review.
11/10/2021	Completed 30-day review, no additional comments.
12/14/2021	Integrated
12/18/2021	Updated the navigation and date for a 2022 release.
12/23/2021	Pre-ratification review update (ZNS: change “change caused,” to “change, caused”)
1/4/2022	Updated text in figure 42.

Description of Changes

NVM Express Base specification:

Add reference to the ISO/IEC 27040 Information Technology - Security Techniques - Storage Security for sanitize operations.

Clarify in the theory of operation that communication between “fabric nodes” is referring to communication between hosts and subsystems.

Add a new section heading to existing text that describes Completion Queue flow control; add references pointing to this new heading.

Remove text that described PCIe Interrupts and move that text to the PCIe Transport specification.

Correct incorporation error for TP4071b that support for the Commands Supported and Effects log page is mandatory.

Correct Administrative controller Admin Command support table use of a footnote that didn't exist (change it to reference the correct footnote); correct incorporation error that the Disconnect command is not permitted on an Administrative controller.

Correct incorporation error that the Disconnect command is not permitted on a Discovery controller.

Correct several misspelled words.

Clarify conditions under which support for Endurance Group Event Aggregate log page is required.

Clarify that “is not used” means “shall be ignored”.

Correct several grammatical errors (missing period at end of sentence, missing commas, single/plural mismatch).

Add reference in the PSDT field pointing to the alignment requirements in bit 17 of the SGLS field.

Clarify that Figures 104, 105, and 106 are examples of single domain NVM subsystems.

The Error Count field in the Error Information Log Entry had conflicting descriptions related to the size of the field and wrap around conditions when the maximum value was reached. One description indicated a 64-bit value that was continually incremented by 1, while the other description indicated that the value wrapped around from FFFFFFFFh (a 32-bit value) to 1. This is a 64-bit value that should wrap around at the appropriate 64-bit value, and not half-way through the range of valid values. This may be viewed as a technical change.

Correct language for returning an error in the status code field to use consistent language throughout.

Correct the figure reference for the timestamp field. The original reference (in rev 1.4) mentioned both the name of the “Get Features” data structure, along with the figure number of the “Set Features” data structure (which are different). Since one figure does not include the timestamp origin information (the referenced figure number) and the other does include the timestamp origin information (the referenced figure name) this may be viewed as a technical change. The figure that included the timestamp origin information was intended to be used.

Add clarification to TREQ field that fabric secure channel includes authentication (to enable searches for the term “authentication” to find this intended case).

Correct case in several section headings.

Apply the same name to all uses of the Copy Format and Copy Descriptor Format field names.

Clarify that the IDLP list of cases is an example list, not a complete list.

Correct uses of LBA Status Information Alert Notices name (several uses did not include the term “Alert”).

Correct reference for the Power Management feature to point to the correct section.

Remove description of a Get Features requirement from the Set Features description of the Host Metadata feature. The Get Features requirements are already included in the Get Features section.

Correct “set to 0” to be “cleared to 0”.

Clarify the description of the ELEN field in the Metadata Element Descriptor.

Correct Controller Metadata references to point to the Host Metadata Feature section rather than the Spinup Control section.

Add reference to CC.AMS in the CATTR field of the Connect command to point to the text that describes how the arbitration mechanism is selected.

Correct reference for the Reservation Acquire command to point to the correct section (7.2 instead of 7.1).

Clarify the description of the Flush command actions.

Correct structuring of the a/b/c list in the ANA effects for Get Log Page commands.

Add heading for Fabric Secure Channel and clarify a failure case related to fabric secure channel.

Clarify that support for the VS UUID capability requires indication of this support in the FID Supported and Effects log page.

NVM Express NVM Command Set:

Clarify some cases related to non-volatile requirements.

Clarify that the PRACT field and the PRCHK field are ignored when end-to-end data protection is not used. Correct references for end-to-end data protection from section 5.1, to correctly point to section 5.2.

Clarify that only the Storage Tag Size (STS) field defines the size of the Storage Tag field.

Clarify that the ELBATM field and the ELBAT field are ignored when end-to-end data protection is not used.

Clarify descriptions of the Copy Command Descriptor Format field to be consistent across the NVM Command Set specification and the Base specification. Update field names so both specifications match when referring to the same field.

Make the implicit requirement for setting bits in the Copy Descriptor Formats Supported field into an explicit requirement.

Ensure the list of commands in the Format Command description that are able to operate with protection information is complete (add Copy, Write Zeroes, and Zone Append).

Update the less precise “smaller” term with the more precise mathematical term “less than”.

Correct reference in OPTPERF description (to point to the corrections sections in the NVM Command Set specification, rather than to the Base Specification).

Correct the name and version of the NVM Command Set specification in the PIIFB bit description in the Identify Namespace Data Structure, NVM Command Set figure.

Clarify the description of the LBSTM field; that not all the bits in this field are required to be used.

Correct the Figure 102 heading to use the proper name of the data structure that is shown.

Update the reference in the Namespace management command to point to the Base Specification rather than an incorrect section in the NVM Command Set specification.

Clarification of Untracked LBAs for the Get LBA Status command. Correct case of several field names to match throughout the document. Clarify the meaning of 0h in the NLSD field. Clarify the meanings of CMPC codes 1h and 2h. Clarify the meaning of the NLB field in the LBA Status Descriptor Entry. Change 2 single bits in the Status field of the LBA Status Descriptor Entry into a 2 bit field to allow for a full description of all possible cases.

Correct grammatical errors.

Update reference for protection information in Copy commands to point to the right section.

Clarify definition of “unmasked” and “masked” in the PRCHK description.

Correct sentence structure for returning error codes to use a common format throughout the document.

Correct uses of LBA Status Information Alert Notices name (several uses did not include the term “Alert”).

Clarify how the RAE bit is used for the Get LBA Status command. Update case for the names of several log pages.

NVM Express Zoned Namespace Command Set:

Clarify that transitions to the Implicitly Opened state or the Full state can occur if any LBAs are written (even if the command that caused the write ultimately returned an error).

Add several additional “refer to” statements to provide pointers for further details.

Clarify that the NSID field in the response to an AER command only applies for a Zoned Namespace Command Set Notice AEN, and not to other AEN events.

A duplicate statement about explicit transition from Read Only state to Offline state was removed from the Annex (the state machine already has this statement).

NVM Express Key Value Command Set:

Implicit requirements related to NUSE and NSZE were reworded / clarified to be explicit statements.

Capitalization of titles was corrected.

Correct a use of the term “index” to be “Format Index”.

Clarified that the term “KV Key Max” referenced the “KV Key Max Length field” and the term “KV Value Max” referenced the “KV Value Max Length field”.

NVM Express PCI Express Transport:

PCI interrupt related descriptions were moved from the NVM Express Base specification into this specification.

NVM Express TCP Transport:

Clarify usage of the Header Digest field (HDGST) and the Data Digest field (DDGST). Each of those fields is present or not present based on the negotiations that occur when the connection is created. This means the offsets for subsequent fields in the PDU change depending on whether these fields are present or not. The text and figures that described this were updated to make this clear.

Error handling related to illegal combinations of HDGSTF flags and DDGSTF flags was clarified.

Editor’s Note:

BLACK text indicates unchanged text; **BLUE** text indicates newly inserted text, **RED** text indicates deleted text; **GREEN** text indicates editor notes.

Description of NVM Express Base specification changes

Modify a portion of section 1.5 as shown below:

1.5.52 sanitize operation

Process by which all user data in the NVM subsystem is altered such that recovery of the previous user data from any cache or the non-volatile media is ~~not possible~~ infeasible for a given level of effort (refer to ISO/IEC 27040).

...

Modify a portion of section 2 as shown below:

2 Theory of Operation

...

There are two defined constructs for communication between the host and the NVM subsystem, a memory-based transport model and a message-based transport model. All NVMe subsystems require the underlying NVMe Transport to provide reliable NVMe command and data delivery. An NVMe Transport is an abstract protocol layer independent of any physical interconnect properties. A taxonomy of NVMe Transports along with examples is shown in Figure 4. An NVMe Transport may expose a memory-based transport model or a message-based transport model. The message-based transport model has two subtypes: the message-only transport model and the message/memory transport model. A memory-based transport model is one in which commands, responses, and data are transferred between a host and an NVM subsystem ~~fabric nodes~~ by performing explicit memory read and write operations. A message-based transport model is one in which messages containing command capsules and response capsules are sent between a host and an NVM subsystem ~~fabric nodes~~. The two subtypes of message-based transport models are differentiated by how data is sent between a host and an NVM subsystem ~~fabric nodes~~. In the message-only transport model data is only sent between a host and an NVM subsystem ~~fabric nodes~~ using capsules or messages. The message/memory-based transport model uses a combination of messages and explicit memory read and write operations to transfer command capsules, response capsules and data between a host and an NVM subsystem ~~fabric nodes~~. Data may optionally be included in command capsules and response capsules. Both the message-only transport model and the message/memory-based transport model are referenced as message-based transport models throughout this specification when the description is applicable to both subtypes.

...

Modify a portion of section 2.2 as shown below:

2.2 Message-Based Transport Model

...

- NVMe over Fabrics does not support Completion Queue flow control (refer to section 3.3.1.2.1). This requires that the host ensures there are available Completion Queue slots before submitting new commands; and

Modify a portion of section 3.1 as shown below:

3.1 NVM Controller Architecture

...

3.1.1 Controller Model

...

In a static controller model, controllers that may be allocated to a particular host may have different state at the time the association is established. The controllers within an NVM subsystem are distinguished by their controller identifier. All memory-based transport model controllers shall support the static controller model.

...

~~NVMe over Fabrics controllers~~ Controllers using the message-based transport model in an NVM subsystem may use a dynamic or static controller model. A Discovery controller shall support the dynamic controller model.

...

~~When using~~ To use the dynamic controller model, the host ~~shall~~ specify a controller identifier of FFFFh when using the Fabrics Connect command (refer to section 6.3) to establish an association with an NVM subsystem.

...

3.1.2 Controller Types

...

Upon completion of the command execution by the NVM subsystem, the controller presents completion queue entries to the host through the appropriate Completion Queues. ~~If MSI-X or multiple message MSI is in use (refer to the Interrupts section of the NVMe over PCIe Transport Specification), then the interrupt vector indicates the Completion Queue(s) with possible new command completions for the host to process. If pin-based interrupts or single message MSI interrupts are used, host software interrogates the Completion Queue(s) for new completion queue entries. The host updates the CQ Head doorbell register to release completion queue entries to the controller and to clear the associated interrupt.~~ Transport specific methods (e.g., PCIe interrupts) are used to notify the host of completion queue entries to process (refer to the appropriate Transport specification).

...

3.1.2.1.2 Log Page Support

...

Figure 24: I/O Controller – Log Page Support

Log Page Name	Log Page Support Requirements ¹
...	
Changed Namespace List	O
Commands Supported and Effects	O M ³
Device Self-test	O
...	
Notes: 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited 2. Mandatory for controllers that support Fixed Capacity Management (refer to section 8.3.2). 3. Optional for NVM Express revision 1.4 and earlier.	

...

3.1.2.2 Administrative Controller

...

~~Since an Administrative controller does not provide access to user data stored on an NVM subsystem's non-volatile storage medium, the~~ An Administrative controller shall not support I/O queues. ~~or namespaces~~ shall not be attached to ~~the~~ an Administrative controller.

...

3.1.2.2.1 Command Support

...

Figure 28: Administrative Controller – Admin Command Support

Command	Command Support Requirements ¹	Reference
...		
Property Set	M ⁶⁵	6.6
Connect	M ⁶⁵	6.3
Property Get	M ⁶⁵	6.5
Authentication Send	O ⁶⁵	6.2
Authentication Receive	O ⁶⁵	6.1
Disconnect	O ⁶ P	6.4
...		
Notes: ... 4. Mandatory if Telemetry Log, Firmware Commit, or SMART/Health Critical Warnings are supported. 5. For NVMe over PCIe implementations, all Fabrics commands are prohibited. For NVMe over Fabrics implementations, the commands are as noted in the table.		

...

3.1.2.3.2 Command Support

...

Figure 32: Discovery Controller – Admin Command Support

Command	Command Support Requirements ¹	Reference
...		
Disconnect	O P	6.4
...		
Notes:		
1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited		
2. For Discovery controllers that do not support explicit persistent connections, the command is reserved prohibited. For Discovery controllers that support explicit persistent connections, the command is mandatory.		

...

3.1.3 Controller Properties

A property is a dword, or qword attribute of a controller. The attribute may have read, write, or read/write access. The host shall access a property using the width specified for that property with an offset that is at the beginning of the property unless otherwise noted in a transport specific specification. All reserved properties and all reserved bits within properties are read-only and return 0h when read. Properties may be read with the Property Get command and may be written with the Property Set command with controllers using the message-based transport model. For controllers using the memory-based transport model, refer to the applicable NVMe Transport binding specification for access methods and rules (e.g., NVMe PCIe ~~Transport~~ Transport Specification).

...

Modify a portion of section 3.2 as shown below:

3.2 NVM Subsystem Entities

...

3.2.3 Endurance Groups

...

If Endurance Groups are supported, then the NVM subsystem and all controllers shall:

- ~~I~~ndicate support for Endurance Groups in the Controller Attributes field in the Identify Controller data structure;
- ~~I~~ndicate the Endurance Group Identifier with which the namespace is associated in the Identify Namespace data structure; ~~and~~
- ~~S~~upport the Endurance Group Information log page; ~~and~~
- support the Endurance Group Event Aggregate log page.

If Endurance Groups are not supported and the host sends a command in which an Endurance Group Identifier field is defined (e.g., Get Log Page), then that field ~~is not used~~ shall be ignored by the controller.

...

Modify a portion of section 3.3 as shown below:

3.3 NVM Queue Models

...

3.3.1.2 Queue Usage

...

Once a submission queue entry or a completion queue entry has been consumed, the slot in which it was placed is free and available for reuse. Altering a submission queue entry after that entry has been submitted but before that entry has been consumed results in undefined behavior. Altering a completion queue entry after that entry has been posted but before that entry has been consumed results in undefined behavior.

3.3.1.2.1 Completion Queue Flow Control

If there are no free slots in a Completion Queue, then the controller shall not post status to that Completion Queue until slots become available. In this case, the controller may stop processing additional submission queue entries associated with the affected Completion Queue until slots become available. The controller shall continue processing for other Submission Queues not associated with the affected Completion Queue.

...

3.3.2 Message-based Transport Queue Model

...

Flow control differs for Submission Queues (refer to section 3.3.2.1.1, section 3.3.2.6, and section 3.3.2.7) and Completion Queues (refer to sections ~~3.3.2.1.1~~ 3.3.2.1.2, ~~3.3.2.6~~, and section 3.3.2.8, and section 3.3.1.2.1).

...

3.3.2.5 Submission Queue Flow Control Negotiation

...

The maximum size of the Admin Submission Queue is specified in the Admin Max SQ Size (ASQSZ) field of the Discovery Log Page Entry for the NVM subsystem (refer to section ~~5.16.1.21~~ 5.16.1.23).

...

3.3.2.8 Completion Queue Considerations

Completion Queue flow control (refer to section 3.3.1.2.1) is not used in the message-based transport queue model. Message-based transport Completion Queues do not use either Head entry pointers or Tail entry pointers.

...

3.3.3.1 Submission Queue Entry

...

Figure 86: Command Dword 0

Bits	Description
...	

Figure 86: Command Dword 0

Bits	Description										
15:14	<p>PRP or SGL for Data Transfer (PSDT): This field specifies whether PRPs or SGLs are used for any data transfer associated with the command. PRPs shall be used for all Admin commands for NVMe over PCIe implementations. SGLs shall be used for all Admin and I/O commands for NVMe over Fabrics implementations (i.e., this field set to 01b). An NVMe Transport may support only specific values (refer to the applicable NVMe Transport binding specification for details).</p> <table> <tr> <th>Value</th><th>Definition</th></tr> <tr> <td>00b</td><td>PRPs are used for this transfer.</td></tr> <tr> <td>01b</td><td> <p>SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of a single contiguous physical buffer that is byte-aligned.</p> <p>Refer to bit 17 of the SGLS field in the Identify Controller data structure (refer to Figure 275) for alignment requirements.</p> </td></tr> <tr> <td>10b</td><td>SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of an SGL segment containing exactly one SGL Descriptor that is qword aligned.</td></tr> <tr> <td>11b</td><td>Reserved</td></tr> </table> <p>If there is metadata that is not interleaved with the user data, as specified in the Format NVM command, then the Metadata Pointer (MPTR) field is used to point to the metadata. The definition of the Metadata Pointer field is dependent on the setting in this field. Refer to the applicable NVMe I/O Command Set specification Figure 87.</p>	Value	Definition	00b	PRPs are used for this transfer.	01b	<p>SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of a single contiguous physical buffer that is byte-aligned.</p> <p>Refer to bit 17 of the SGLS field in the Identify Controller data structure (refer to Figure 275) for alignment requirements.</p>	10b	SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of an SGL segment containing exactly one SGL Descriptor that is qword aligned.	11b	Reserved
Value	Definition										
00b	PRPs are used for this transfer.										
01b	<p>SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of a single contiguous physical buffer that is byte-aligned.</p> <p>Refer to bit 17 of the SGLS field in the Identify Controller data structure (refer to Figure 275) for alignment requirements.</p>										
10b	SGLs are used for this transfer. If used, Metadata Pointer (MPTR) contains an address of an SGL segment containing exactly one SGL Descriptor that is qword aligned.										
11b	Reserved										
...											

...

Modify a portion of section 3.4 as shown below:

3.4.2 Fused Operations

Fused operations enable a more complex command by “fusing” together two simpler commands. This feature is optional; support for this feature is indicated in [the FUSES field](#) in the Identify Controller data structure in Figure 275. In a fused operation, the requirements are:

...

Modify a portion of section 3.7 as shown below:

<Editor’s note: The change in this section adds the period at the end of each sentence.>

3.7.2 Controller Level Reset

The following methods initiate a Controller Level Reset:

- ...
- Transport specific reset types (refer to the applicable NVMe Transport binding specification), if any.

A Controller Level Reset consists of the following actions:

- ...
- All controller properties defined in section 3.1.3 and internal controller state are reset, with the following exceptions:

- for controllers using a memory-based transport:
 - ...
- for controllers using a message-based transport:
 - there are no exceptions.

...

Modify a portion of section 3.8 as shown below:

3.8 NVM Capacity Model

...

3.8.2.1 Simple NVM Subsystem

~~In the example shown in~~ Figure 104, ~~shows an example of a single domain NVM subsystem where~~ endurance is managed across all media units.

...

3.8.2.2 Vertically-Organized NVM Subsystem

~~In the example shown in~~ Figure 105, ~~shows an example of a single domain NVM subsystem where~~ the performance goal is isolation among four NVM Sets at the cost of bandwidth.

...

3.8.2.3 Horizontally-Organized Dual NAND NVM Subsystem

~~In the example shown in~~ Figure 106, ~~shows an example of a single domain NVM subsystem where~~ the Media Units are NAND which is capable of being operated as QLC or at a lower density.

...

3.8.3 Capacity Reporting

...

Figure 107: Capacity Information Field Usage

Entity being created / deleted	NVM Sets supported	Endurance Groups supported	Domains supported	Capacity information used ¹
...				
NOTES: ... 5. Capacity information in the Endurance Group Information log page (i.e., TEGCAP field, UEGCAP field s (refer to Figure 217)). ...				

...

Modify a portion of section 4.1 as shown below:

4.1 Data Layout

...

4.1.1 Physical Region Page Entry and List

...

A physical region page list (PRP List) is a set of PRP entries in a single page of contiguous memory. A PRP List describes additional PRP entries that could not be described within the command itself. Any PRP entries described within the command are not duplicated in a PRP List. If the amount of data to transfer requires multiple PRP List memory pages, then the last PRP entry before the end of the memory page shall be a pointer to the next PRP List, indicating the next segment of the PRP List. Figure 110 shows the layout of a PRP List where each PRP entry identifies memory pages that are physically contiguous. Figure 111 shows the layout of a PRP List where each PRP entry identifies a different memory page (i.e., the memory pages are not physically ~~contiguous~~ contiguous).

...

Modify a portion of section 5.16 as shown below:

5.16 Get Log Page command

...

5.16.1.2 Error Information (Log Identifier 01h)

...

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

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

...

5.16.1.8 Telemetry Host-Initiated (Log Identifier 07h)

This log consists of a header describing the log and zero or more Telemetry Data Blocks (refer to section 8.24). All Telemetry Data Blocks are 512 bytes in size. The controller shall initiate a capture of the controller's internal controller state to this log if the controller processes a Get Log Page command for this log with the Create Telemetry Host-Initiated Data bit set to '1' in the Log Specific field. If the host specifies a Log Page Offset Lower value that is not a multiple of 512 bytes in the Get Log Page command for this log, then the controller shall ~~return an error~~ abort the command with a status code ~~set to~~ of Invalid Field in Command. This log page is global to the controller or global to the NVM subsystem.

...

5.16.1.14 Persistent Event Log (Log Identifier 0Dh)

...

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

Bytes	Description
Persistent Event Log Header	
...	
27:20	Timestamp: Shall contain a timestamp of the time at which the persistent event log reporting context was established. The value returned shall use using the Timestamp data structure defined in Figure 339 Figure 340.
...	

...

5.16.1.23 Discovery Log Page (Log Identifier 70h)

...

Figure 264: Get Log Page – Discovery Log Page Entry

Bytes	Description										
...											
03	<p>Transport Requirements (TREQ): Indicates requirements for the NVMe Transport.</p> <p>Bits 7:3 are reserved.</p> <p>Bit 2 if set to '1' indicates that the controller is capable of disabling SQ flow control. A controller that is capable of disabling SQ flow control may accept or reject a host request to disable SQ flow control. If cleared to '0', then the controller requires use of SQ flow control.</p> <p>Bits 1:0 indicate whether connections shall be made over a fabric secure channel (which includes authentication) (refer to section 8.13).</p> <table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Not specified</td></tr><tr><td>01b</td><td>Required</td></tr><tr><td>10b</td><td>Not required</td></tr><tr><td>11b</td><td>Reserved</td></tr></table>	Value	Definition	00b	Not specified	01b	Required	10b	Not required	11b	Reserved
Value	Definition										
00b	Not specified										
01b	Required										
10b	Not required										
11b	Reserved										
...											

...

Modify a portion of section 5.17 as shown below:

5.17 Identify command

...

5.17.2.1 Identify Controller ~~d~~Data ~~s~~Structure (CNS 01h)

...

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

Bytes	I/O ¹	Admin ¹	Disc ¹	Description								
Controller Capabilities and Features												
...												
95:92	M	M	M	Optional Asynchronous Events Supported (OAES): ... Bit 13 is set to '1' if the controller supports the LBA Status Information Alert Notices event (refer to the NVM Command Set specification). If cleared to '0', then the controller does not support the LBA Status Information Alert Notices event. ...								
...												
535:534	M	R	R	Optional Copy Descriptor Formats Supported: <table><tr><th>Bits</th><th>Description</th></tr><tr><td>15:2</td><td>Reserved</td></tr><tr><td>1</td><td>If set to '1', then the controller supports Copy Descriptor Format 1h. If cleared to '0', then the controller does not support Copy Descriptor Format 1h.</td></tr><tr><td>0</td><td>If set to '1', then the controller supports Copy Descriptor Format 0h. If cleared to '0', then the controller does not support Copy Descriptor Format 0h.</td></tr></table>	Bits	Description	15:2	Reserved	1	If set to '1', then the controller supports Copy Descriptor Format 1h. If cleared to '0', then the controller does not support Copy Descriptor Format 1h.	0	If set to '1', then the controller supports Copy Descriptor Format 0h. If cleared to '0', then the controller does not support Copy Descriptor Format 0h.
Bits	Description											
15:2	Reserved											
1	If set to '1', then the controller supports Copy Descriptor Format 1h. If cleared to '0', then the controller does not support Copy Descriptor Format 1h.											
0	If set to '1', then the controller supports Copy Descriptor Format 0h. If cleared to '0', then the controller does not support Copy Descriptor Format 0h.											
...												

...

Figure 276: Identify – Power State Descriptor Data Structure

Bits	Description
...	
143:128	Idle Power (IDL P): This field indicates the typical power consumed by the NVM subsystem over 30 seconds in this power state when idle (l e.g., there are no pending commands, property accesses, background processes, sanitize operation, nor device self-test operations). The measurement starts after the NVM subsystem has been idle for 10 seconds. ...
...	

...

Modify a portion of section 5.24 as shown below:

<Editor's note: The change in this section replaces the period in the middle of the sentence with a comma.>

5.24 Sanitize command

...

If the Firmware Commit command that established the pending firmware activation with reset condition returned a status code of:

- Firmware Activation Requires Controller Level Reset;
- Firmware Activation Requires Conventional Reset; or
- Firmware Activation Requires NVM Subsystem Reset,

then the controller should abort the Sanitize command with that same status code.

...

Modify a portion of section 5.27 as shown below:

5.27.1.8 Asynchronous Event Configuration (Feature Identifier 0Bh)

...

Figure 326: Asynchronous Event Configuration – Command Dword 11

Bits	Description
...	
13	LBA Status Information Alert Notices ¹ : I/O Command Set specific definition.
...	

...

5.27.1.9 Autonomous Power State Transition (Feature Identifier 0Ch), (Optional)

...

Figure 329: Interactions between APSTE and NOPPME

APSTE ¹	NOPPME ²	Non-operational power state entry	Background operations during non-operational power states
1	1	Entered by host request ³ or by ITPT idle timer ⁴	Allowed
0	1	Entered by host request ³	Allowed
1	0	Entered by host request ³ or by ITPT idle timer ⁴	Not allowed
0	0	Entered by host request ³	Not allowed
NOTES: 1. Defined in Figure 327. 2. Defined in Figure 343. 3. Refer to section 5.27.1.1 5.27.1.2. 4. Refer to Figure 328.			

...

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

...

Figure 359: Set Features – Command Dword 11

Bits	Description
31:15	Reserved

Figure 359: Set Features – Command Dword 11

Bits	Description										
14:13	Element Action (EA): This field specifies the action to perform on the specified Host Metadata Feature value for each Metadata Element Descriptor data structure contained in the Host Metadata data structure. This field shall be cleared to 0h for a Get Features command.										
	<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Add/Replace Entry</td></tr><tr><td>01b</td><td>Delete Entry Multiple</td></tr><tr><td>10b</td><td>Add Entry Multiple</td></tr><tr><td>11b</td><td>Reserved</td></tr></table>	Value	Definition	00b	Add/Replace Entry	01b	Delete Entry Multiple	10b	Add Entry Multiple	11b	Reserved
	Value	Definition									
	00b	Add/Replace Entry									
	01b	Delete Entry Multiple									
10b	Add Entry Multiple										
11b	Reserved										
If the Element Action field is cleared to 00b (Add/Replace Entry) and the Metadata Element Descriptor with the specified Element Type (refer to Figure 361) does not exist in the specified Host Metadata Feature value, then the Controller shall create the descriptor in the specified Host Metadata Feature value with the value in the Host Metadata data structure.											
If the Element Action field is set cleared to 00b (Add/Replace Entry) and one Metadata Element Descriptor with the specified Element Type exists in the specified Host Metadata Feature value, then the Controller shall replace with the value in the specified Host Metadata data structure.											
If the Element Action field is cleared to 00b (Add/Replace Entry) and the Feature Identifier field is set to Enhanced Controller Metadata, then the controller shall abort the Set Features command with a status code of Invalid Field in Command and shall not change any Host Metadata Feature value.											
...											
12:00	Reserved										

...

Figure 361: Metadata Element Descriptor

Bit	Description
31 + (Element Length*8) :32	Element Value (EVAL): This field specifies the value for the element.
31:16	Element Length (ELEN): This field specifies the length of the Element Value field in bytes. This field shall be cleared to 0h when deleting an entry (i.e., the EA = field is set to 01b in Command Dword 11). This field should be non-zero when adding/updating and an entry (i.e., the EA = field is cleared to 00b). If this field is cleared to 0h when adding/updating an entry, then the controller behavior is undefined.
...	

...

5.27.1.23.1 Enhanced Controller Metadata (Feature Identifier 7Dh)

...

Refer to section ~~5.27.1.22~~ **5.27.1.23** for the definitions of Command Dword 11 and the Host Metadata Data Structure.

...

5.27.1.23.2 Controller Metadata (Feature Identifier 7Eh)

...

Refer to section ~~5.27.1.22~~ **5.27.1.23** for the definitions of Command Dword 11 and the Host Metadata Data Structure.

...

5.27.1.23.3 Namespace Metadata (Feature Identifier 7Fh)

...

Refer to section ~~5.27.1.22~~ 5.27.1.23 for the definitions of Command Dword 11 and the Host Metadata Data Structure.

...

Modify a portion of section 5.28 as shown below:

5.28 Virtualization Management command

...

If the action requested specifies a range of controller resources that:

- a) does not exist;
- b) is a Private Resource (e.g., VQ resources are requested when VQ resources are not supported, VI resources are requested when VI resources are not supported); or
- c) is currently in use (e.g., the number of Controller Resources (NR) is greater than the number of remaining available flexible resources),

then ~~an~~ the command is aborted with a status code of Invalid Resource Identifier ~~is returned~~.

...

Modify a portion of section 6.3 as shown below:

6.3 Connect Command and Response

...

Figure 380: Connect Command – Submission Queue Entry

Bytes	Description									
...										
46	Connect Attributes (CATTR): This field indicates attributes for the connection.									
	...									
	Bits 1:0 indicate the priority class to use for commands within this Submission Queue. This field is only used when the weighted round robin with urgent priority class is the arbitration mechanism selected (refer to CC.AMS in Figure 46), the field is ignored if weighted round robin with urgent priority class is not used. Refer to section 3.4.4. This field is only valid for I/O Queues. It and shall be cleared to 00b for Admin Queue connections.									
	<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Urgent</td></tr><tr><td>01b</td><td>High</td></tr><tr><td>10b</td><td>Medium</td></tr><tr><td>11b</td><td>Low</td></tr></table>	Value	Definition	00b	Urgent	01b	High	10b	Medium	11b
Value	Definition									
00b	Urgent									
01b	High									
10b	Medium									
11b	Low									
...										

...

Modify a portion of section 7 as shown below:

7 I/O Commands

...

Figure 390: Opcodes for I/O Commands

Opcode by Field			Combined Opcode ¹	Command ²	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer ³			
...					
0b	000 11b	10b	0Eh	Reservation Report	7.5
0b	001 00b	01b	11h	Reservation Acquire	7.1 7.2
0b	001 01b	01b	15h	Reservation Release	7.4
Vendor Specific					
...					

...

Modify a portion of section 7.1 as shown below:

7.1 Flush command

...

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

- a) shall complete successfully ~~and have no effect~~ if a sanitize operation is not in progress; and
- b) may complete successfully ~~and have no effect~~ if a sanitize operation is in progress.

Modify a portion of section 8.1 as shown below:

8.1 Asymmetric Namespace Access Reporting

...

8.1.4 Asymmetric Namespace Access States Command Processing Effects

...

Figure 411: ANA effects on Command Processing

Command	ANA State	Effects on command processing
...		

Figure 411: ANA effects on Command Processing

Command	ANA State	Effects on command processing
Get Log Page	ANA Inaccessible, ANA Persistent Loss, or ANA Change	<p>The following log pages are affected:</p> <ul style="list-style-type: none"> a) Error Information (i.e., 01h): The log page is not required to contain entries for namespaces whose relationship to the controller processing the command is in the: <ul style="list-style-type: none"> a. ANA Inaccessible state (refer to section 8.1.3.3); b. the ANA Persistent Loss state (refer to section 8.1.3.4); or c. the ANA Change state (refer to section 8.1.3.5). <p>The following log pages are not available¹:</p> <ul style="list-style-type: none"> a) Media Unit Status log page (refer to section 5.16.1.16); and b) Supported Capacity Configuration List log page (refer to section 5.16.1.17).
<p>NOTES:</p> <p>1. If the ANA state is ANA Inaccessible State, then commands that use feature identifiers or log pages that are not available shall abort with a status code of Asymmetric Access Inaccessible. If the ANA state is ANA Persistent Loss State, then commands that use feature identifiers or log pages that are not available shall abort with a status code of Asymmetric Access Persistent Loss. If the ANA state is ANA Change State, then commands that use feature identifiers or log pages that are not available shall abort with a status code of Asymmetric Access Transition.</p> <p>...</p>		

...

Modify a portion of section 8.13 as shown below:

8.13 NVMe over Fabrics **Secure Channel and In-band Authentication**

NVMe over Fabrics supports both fabric secure channel (that includes authentication) and NVMe in-band authentication.

...

8.13.1 Fabric Secure Channel

...

If establishment of a secure channel fails or a secure channel is not established when required by the controller, the resulting errors are fabric-specific and may not be reported to the NVMe layer on the host. Such errors may result in the controller being inaccessible to the host via the NVMe Transport connection on which the ~~authentication~~ failure to establish a fabric secure channel occurred.

...

Modify a portion of section 8.15 as shown below:

8.15 Power Management

...

The Idle Power (IDLP) field indicates the typical power consumed by the NVM subsystem over 30 seconds in the power state when idle (i.e.g., there are no pending commands, property accesses, background processes, nor device self-test operations). The measurement starts after the NVM subsystem has been idle for 10 seconds.

Modify a portion of section 8.25 as shown below:

<Editor's note: Correct all lists in the section 8.25 to be real Word lists, rather than plain text.>

8.25 Universally Unique Identifiers (UUIDs) for Vendor Specific Information

...

8.25.3.2 Vendor Specific Feature Example

...

A controller supporting both definitions of the feature for the Get Features command:

- a) Sets the UUID List bit to '1' in the CTRATT field of the Identify Controller data structure (refer to Figure 275);
- b) Sets the UUID Selection Supported bit to '1' in the Commands Supported and Effects data structure (refer to Figure 211) corresponding to the Get Features command; ~~and~~
- c) ~~Sets the UUID Selection Supported bit to '1' in the FID Supported and Effects log page (refer to Figure 256); and~~
- d) Reports both UUID V and UUID C in the UUID list (refer to Figure 284).

...

Description of NVM Express NVM Command Set specification changes

Modify a portion of section 2.1 as shown below:

2.1 Theory of operation

...

2.1.4.2.2 Non-volatile requirements

After a write command has completed **without error**, reads for that location which are subsequently submitted **and return data**, shall return the data ~~from that was written by~~ that write command and not an older version of the data from previous write commands with the following exception:

If all of the following conditions are met:

- a) the controller supports a volatile write cache;
- b) the volatile write cache is enabled;
- c) the FUA bit for the write is not set;
- d) no flush commands, associated with the same namespace as the write, successfully completed before the controller reports shutdown complete (CSTS.SHST set to 10b); and
- e) main power loss occurs on a controller without completing the normal or abrupt shutdown procedure outlined in the Memory-based Transport Controller Shutdown or Message-based Transport Controller Shutdown sections in the NVMe Base Specification,

then subsequent reads for locations written to the volatile write cache that were not written to non-volatile storage may return older data.

...

2.1.5 End-to-end Protection Information

...

Figure 9: Protection Information Field Definition

Bits	Description
03	Protection Information Action (PRACT): This bit indicates specifies the action to take for the protection information. If the namespace is not formatted to use end-to-end protection information, then this bit is shall be ignored by the controller . Refer to section 5.4 5.2
02:00	Protection Information Check (PRCHK): The protection information check field specifies the fields that shall be checked as part of end-to-end data protection processing. If the namespace is not formatted to use end-to-end protection information, then this field is shall be ignored by the controller . Refer to section 5.4 5.2

Figure 10: Storage Tag Check Definition

Bits	Description
00	<p>Storage Tag Check: This bit specifies the checking requirements for the Storage Tag field, if defined, shall be checked as part of end-to-end data protection processing. If set to '1', then protection information checking of the Storage Tag field is enabled. If cleared to '0', the Storage Tag field is not checked. Refer to section 5.4 5.2.</p> <p>If:</p> <ul style="list-style-type: none"> a) the namespace is formatted to use end-to-end protection; b) the protection information format is (refer to section 5.2.1): <ul style="list-style-type: none"> ○ 16b Guard Protection Information; or ○ 64b Guard Protection Information; <p>and</p> <p>c)</p> <p>If the Storage Tag Size (STS) field is cleared to 0h (refer to Figure 101), then this bit is shall be ignored by the controller as no Storage Tag field is defined.</p>

2.1.6 Metadata Region (MR)

...

If the namespace is formatted to use end-to-end data protection (refer to section 5.4 5.2), then the first bytes or last bytes of the metadata is used for protection information (specified as part of the Format NVM command).

...

Modify a portion of section 3.2 as shown below:

3.2 NVM Command Set Commands

...

3.2.1 Compare command

...

Figure 25: Compare – Command Dword 15

Bits	Description
31:16	Expected Logical Block Application Tag Mask (ELBATM): This field specifies the Application Tag Mask expected value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller. Refer to section 5.4 5.2.
15:00	Expected Logical Block Application Tag (ELBAT): This field specifies the Application Tag expected value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller. Refer to section 5.4 5.2.

...

3.2.2 Copy command

...

Figure 30: Copy – Command Dword 12

Bits	Description		
...			
11:08	Descriptor Format: Specifies the type format of the Source Range Entries as follows: Copy Descriptor Format that is used. The Copy Descriptor Format specifies the starting LBA, number of logical blocks, and parameters associated with the read portion of the operation.		
	Code Copy Descriptor Format type	Description	Reference
	0h	The Source Range Entries Descriptor Format 0h is used (refer to Figure 34) specify starting LBA, number of logical blocks, and parameters associated with the read portion of the operation.	Figure 34
	1h	The Source Range Entries Descriptor Format 1h is used (refer to Figure 35) specify starting LBA, number of logical blocks, and parameters associated with the read portion of the operation when PIF1 bit in the DPC field (refer to Figure) is set to '1'.	Figure 35
	All Others	Reserved	
...			

...

Figure 33: Copy – Command Dword 15

Bits	Description
31:16	Logical Block Application Tag Mask (LBATM): This field specifies the Application Tag Mask value for the write portion of the copy operation. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2.
15:00	Logical Block Application Tag (LBAT): This field specifies the Application Tag value for the write portion of the copy operation. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2.

The controller shall indicate the Source Range Entries Descriptor formats supported by the controller in the Copy Descriptor Formats Supported field in the Identify Controller data structure (refer to the NVMe Base Specification).

The data that the Copy command provides is a list of Source Range Entries that describe the data to be copied to the destination range starting at the SDLBA. The **format Copy Descriptor Format type** of the Source Range Entries is specified in the Descriptor Format field in **Command Dword 12**. If the **Copy Descriptor Format** specified in the Descriptor Format field is not supported by the controller, then the command shall be aborted with a status code of Invalid Field in Command.

If:

- the **Copy Descriptor Format** specified in the Descriptor Format field is supported by the controller;
- the namespace specified by NSID is formatted to use 16b Guard Protection Information; and
- the Descriptor Format is not cleared to 0h,

then the command shall be aborted with the status code of Invalid Namespace or Format.

If:

- the **Copy Descriptor Format** specified in the Descriptor Format field is supported by the controller;
- the namespace specified by NSID is formatted to use 32b Guard Protection Information or 64b Guard Protection Information; and
- the Descriptor Format is not set to 1h,

then the command shall be aborted with the status code of Invalid Namespace or Format.

Figure 34 shows the [Copy Descriptor Format 0h](#) descriptor [format](#) and an example with 128 Source Range entries.

Figure 34: Copy – Source Range Entries Descriptor Format 0h

Range	Bytes	Description
Source Range 0	...	
	27:24	This field specifies the variable sized Expected Logical Block Storage Tag (ELBST) and Expected Initial Logical Block Reference Tag (EILBRT), which are defined in section 5.2.1.4.1, to be used for the read portion of the copy operation for the LBAs specified in this Source Range entry. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
	29:28	Expected Logical Block Application Tag (ELBAT): This field specifies the Application Tag expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
	31:30	Expected Logical Block Application Tag Mask (ELBATM): This field specifies the Application Tag Mask expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .

...

Figure 35 shows the [Copy Descriptor Format 01h](#) descriptor [format](#) and an example with 102 Source Range entries.

Figure 35: Copy – Source Range Entries Descriptor Format 1h

Range	Bytes	Description
Source Range 0	...	
	35:26	This field specifies variable sized Expected Logical Block Storage Tag (ELBST) and Expected Initial Logical Block Reference Tag (EILBRT) fields, which are defined in section 5.2.1.4.1, to be used for the read portion of the copy operation. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.2 .
	37:36	Expected Logical Block Application Tag (ELBAT): This field specifies the Application Tag expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
	39:38	Expected Logical Block Application Tag Mask (ELBATM): This field specifies the Application Tag Mask expected value used for the read portion of the copy operation for the LBAs specified in this Source Range entry. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .

...

3.2.4 Read command

...

Figure 51: Read – Command Dword 15

Bits	Description
31:16	Expected Logical Block Application Tag Mask (ELBATM): This field specifies the Application Tag Mask expected value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
15:00	Expected Logical Block Application Tag (ELBAT): This field specifies the Application Tag expected value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .

...

3.2.5 Verify command

...

Figure 57: Verify – Command Dword 15

Bits	Description
31:16	Expected Logical Block Application Tag Mask (ELBATM): This field specifies the Application Tag Mask expected value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
15:00	Expected Logical Block Application Tag (ELBAT): This field specifies the Application Tag expected value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .

...

3.2.6 Write command

...

Figure 66: Write – Command Dword 15

Bits	Description
31:16	Logical Block Application Tag Mask (LBATM): This field specifies the Application Tag Mask value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
15:00	Logical Block Application Tag (LBAT): This field specifies the Application Tag value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .

...

3.2.8 Write Zeroes command

...

Figure 75: Write Zeroes – Command Dword 15

Bits	Description
31:16	Logical Block Application Tag Mask (LBATM): This field indicates the Application Tag Mask value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .
15:00	Logical Block Application Tag (LBAT): This field indicates the Application Tag value. If the namespace is not formatted to use end-to-end protection information, then this field is ignored by the controller . Refer to section 5.4 5.2 .

Modify a portion of section 4.1 as shown below:

4.1 Admin Command behavior for the NVM Command Set

...

4.1.2 Format NVM command

...

Figure 78: Format NVM – Command Dword 10 – NVM Command Set Specific Fields

Bits	Description
08	Protection Information Location (PIL): If set to '1' and protection information is enabled (refer to section 5.4 5.2), then protection information is transferred as the first bytes of metadata. If cleared to '0' and protection information is enabled, then protection information is transferred as the last bytes of metadata. ...

Figure 78: Format NVM – Command Dword 10 – NVM Command Set Specific Fields

Bits	Description
07:05	<p>Protection Information (PI): This field specifies whether end-to-end data protection is to be enabled and if enabled, the type of protection information to use. The values for this field have the following meanings:</p> <p>...</p> <p>When If end-to-end data protected is enabled, the host shall specifies the appropriate protection information in the Copy commands, Read commands, Verify commands, Write commands, Write Zeroes commands, or and Compare commands.</p>
04	<p>Metadata Settings (MSET): This bit is set to '1' if the metadata is transferred as part of an extended data LBA. This bit is cleared to '0' if the metadata is transferred as part of a separate buffer. The metadata may include protection information, based on the Protection Information (PI) field. If the Metadata Size for the LBA Format selected is 0h, then this bit is not applicable shall be ignored by the controller.</p>

...

4.1.3 Get Features & Set Features commands

...

Figure 79: Feature Identifiers – NVM Command Set

Feature Identifier	Persistent Across Power Cycle and Reset ^{a) 1}	Uses Memory Buffer for Attributes	Description
...			
<p>NOTES:</p> <p>^{a) 1} This column is only valid if the feature is not saveable (refer to the NVMe Base specification). If the feature is saveable, then this column is not used and any feature may be configured to be saved across power cycles and reset.</p>			

...

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

...

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

Bytes	O/M ¹	Description
...		
23:16	M	<p>Namespace Utilization (NUSE): This field indicates the current number of logical blocks allocated in the namespace. This field is smaller less than or equal to the Namespace Capacity. The number of logical blocks is based on the formatted LBA size.</p> <p>Refer to section 2.1.1 for details on the usage of this field.</p>
24	M	<p>Namespace Features (NSFEAT): This field defines features of the namespace.</p> <p>Bits 7:5 are reserved.</p> <p>Bit 4 (OPTPERF) if set to '1' indicates that the fields NPWG, NPWA, NPDG, NPDA, and NOWS are defined for this namespace and should be used by the host for I/O optimization (refer to the NVM Set List section 5.8.2 in the NVMe Base Specification). If cleared to '0', then the controller does not support the fields NPWG, NPWA, NPDG, NPDA, and NOWS for this namespace.</p> <p>...</p>
...		

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

Bytes	O/M ¹	Description							
28	M	End-to-end Data Protection Capabilities (DPC): This field indicates the capabilities for the end-to-end data protection feature. Multiple bits may be set in this field. Refer to section 5.4 5.2.							
		Bits	Description	...		3	Protection Information In First Bytes (PIIFB): 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 revision 1.0 1.4 of the NVMe Base Specification or later of the NVM Command Set Specification , this bit shall be cleared to '0'.	...	
		Bits	Description						
		...							
3	Protection Information In First Bytes (PIIFB): 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 revision 1.0 1.4 of the NVMe Base Specification or later of the NVM Command Set Specification , this bit shall be cleared to '0'.								
...									
29	M	End-to-end Data Protection Type Settings (DPS): This field indicates the protection information Type settings for the end-to-end data protection feature. Refer to section 5.4 5.2.							
...		...							
...									
75:74	O	Maximum Single Source Range Length (MSSRL): This field indicates the maximum number of logical blocks that may be specified in the Number of Logical Blocks field in each valid Source Range Entries Descriptor field (refer to 3) of a Copy command (refer to section 3.2.2). If the controller supports the Copy command, then this field shall be set to a non-zero value.							
...									

...

4.1.5.3 I/O Command Set Specific Identify Namespace Data Structure (CNS 05h)

...

Figure 100: NVM Command Set I/O Command Set Specific Identify Namespace Data Structure (CSI 00h)

Bytes	O/M ¹	Description
7:0	O	<p>Logical Block Storage Tag Mask (LBSTM): Identifies Indicates the mask for the Storage Tag field for the protection information (refer to section 5.4 5.2). The size of the mask contained in this field is defined by the STS field (refer to Figure 101). If the size of the mask contained in this field is less than 64 bits, then the mask is contained in the least-significant bits of this field. The host should ignore bits in this field that are not part of the mask.</p> <p>If end-to-end protection is not enabled in the namespace, then this field is should be ignored by the host.</p> <p>If:</p> <ul style="list-style-type: none"> a) end-to-end protection is enabled; b) 16b Guard Protection Information format is used; and c) the 16BPISTM bit is set to '1' in the PIC field, <p>then all each bits in the mask in this field shall be set to '1'.</p>
...		

...

4.1.5.4 I/O Command Set Specific Identify Controller ~~d~~Data ~~s~~Structure (CNS 06h, CSI 00h)

...

Figure 102: ~~Identify—Identify Controller Data Structure, I/O Command Set Specific Identify Controller Data Structure for the NVM Command Set (CSI-00h)~~

Bytes	O/M ¹	Description
...		

...

4.1.6 Namespace Management command

...

Figure 105: Namespace Management – Host Software Specified Fields

Bytes	Description	Host Specified
...		
Notes:		
1. A value of 0h specifies that the controller determines the value to use (refer to the Namespace Management section of the NVMe Base Specification 8-42). If the associated feature is not supported, then this field is ignored by the controller.		

...

Modify a portion of section 4.2 as shown below:

4.2 I/O Command Set Specific Admin commands

...

4.2.1 Get LBA Status command

...

A controller identifies Potentially Unrecoverable LBAs using the following two report types:

- Tracked LBAs:** a list of Potentially Unrecoverable LBAs associated with physical storage. These may be discovered through a background scan where the controller examines the media in the background or discovered through other means. The Tracked LBA list is able to be returned without significant delay; or
- Untracked LBAs:** a list of Potentially Unrecoverable LBAs generated by a scan originated by a Get LBA Status command with the ATYPE field set to 10h. The controller scans internal data structures related to [the specified range of](#) LBAs to determine which LBAs are Potentially Unrecoverable LBAs. The controller may use this scan to determine which LBAs in which namespaces are affected by a component (e.g., die or channel) failure. Significant delays may be incurred during the processing of a Get LBA Status command with the ATYPE field set to 10h. After discovery of Untracked LBAs, they may or may not be added to the list of Tracked LBAs.

In response to a Get LBA Status command, the controller ~~returns shall return~~ LBA Status Descriptors that describe LBAs written by a Write Uncorrectable command in addition to any other LBAs that may return an Unrecovered Read Error status discovered through other mechanisms. The list of Tracked LBAs and the list of Untracked LBAs may [be included in](#) LBA Status Descriptor Entries that describe LBAs written by a Write Uncorrectable command. If an LBA Status Descriptor Entry describes [only](#) LBAs written by a Write

Uncorrectable command, then bits 1:0 in the Status field **should be set to 11b** in ~~of~~ that entry ~~should be set to '1'.~~

...

<Editor's note: Correct the lists in this section to be real Word a/b/c lists, rather than plain text.>

If the value in the Action Type (ATYPE) field is **set to 10h**, then:

...

- a) the controller shall generate a list of Untracked LBAs as described in this section;
- b) the controller shall return Untracked LBAs and Tracked LBAs in the range specified in the Get LBA Status command for the namespace specified in the Namespace Identifier (CDW1.NSID);

...

- ~~d)~~ d) the controller shall ensure that any such successfully re-written logical blocks are not reported in any LBA Status Descriptor ~~e~~Entries returned by the Get LBA Status command unless, after having been removed from relevant internal data structures and prior to processing the Get LBA Status command, **those LBAs** were newly detected as being Potentially Unrecoverable LBAs; and
- ~~e)~~ e) the list of Untracked LBAs returned by the Get LBA Status command may be discarded by the controller or added to the Tracked LBA list once the command has completed.

...

If the value in the Action Type (ATYPE) field is **set to 11h**, then the controller shall:

...

- c) ensure that any such successfully re-written logical blocks are not reported in any LBA Status Descriptor ~~e~~Entries returned by the Get LBA Status command unless, after having been removed from relevant internal data structures and prior to processing the Get LBA Status command, **those LBAs** were newly detected as being Potentially Unrecoverable LBAs; and

...

Figure 111: LBA Status Descriptor List

Bytes	Description
03:00	Number of LBA Status Descriptors (NLSD): This field indicates the number of LBA Status Descriptor e Entries returned by the controller in this data structure. An LBA Status Descriptor List which indicates that no LBA Status Descriptor entries have been returned (i.e., NLSD is cleared to '0') is a valid LBA Status Descriptor List. A value of 0h in this field indicates that no LBA Status Descriptor Entries are returned.

Figure 111: LBA Status Descriptor List

Bytes	Description										
04	<p>Completion Condition (CMPC): If the command completes successfully, then tThis field indicates the condition that caused completion of the Get LBA Status command. If there are no more LBA Status Descriptor Entries to transfer in the specified range, the controller shall return Completion Condition 2h.</p> <table> <tr> <th>Code</th><th>Definition</th></tr> <tr> <td>0h</td><td>No indication of the completion condition.</td></tr> <tr> <td>1h</td><td> <p>INCOMPLETE: The command completed due to as a result of transferring the amount of data number of Dwords specified in the MNDW field and: There may be more</p> <ul style="list-style-type: none"> for ATYPE set to 10h or ATYPE set to 11h, additional LBA Status Descriptor Entries are available to transfer in that are associated with the specified LBA range; or for ATYPE set to 10h, the scan did not complete. </td></tr> <tr> <td>2h</td><td> <p>COMPLETE: The command completed due to as a result of completing having performed the action specified in the Action Type field over the number of logical blocks specified in the Range Length field. There and there are no more additional LBA Status Descriptor Entries available to transfer in that are associated with the specified range.</p> </td></tr> <tr> <td>All others</td><td>Reserved</td></tr> </table>	Code	Definition	0h	No indication of the completion condition.	1h	<p>INCOMPLETE: The command completed due to as a result of transferring the amount of data number of Dwords specified in the MNDW field and: There may be more</p> <ul style="list-style-type: none"> for ATYPE set to 10h or ATYPE set to 11h, additional LBA Status Descriptor Entries are available to transfer in that are associated with the specified LBA range; or for ATYPE set to 10h, the scan did not complete. 	2h	<p>COMPLETE: The command completed due to as a result of completing having performed the action specified in the Action Type field over the number of logical blocks specified in the Range Length field. There and there are no more additional LBA Status Descriptor Entries available to transfer in that are associated with the specified range.</p>	All others	Reserved
Code	Definition										
0h	No indication of the completion condition.										
1h	<p>INCOMPLETE: The command completed due to as a result of transferring the amount of data number of Dwords specified in the MNDW field and: There may be more</p> <ul style="list-style-type: none"> for ATYPE set to 10h or ATYPE set to 11h, additional LBA Status Descriptor Entries are available to transfer in that are associated with the specified LBA range; or for ATYPE set to 10h, the scan did not complete. 										
2h	<p>COMPLETE: The command completed due to as a result of completing having performed the action specified in the Action Type field over the number of logical blocks specified in the Range Length field. There and there are no more additional LBA Status Descriptor Entries available to transfer in that are associated with the specified range.</p>										
All others	Reserved										
07:05	Reserved										
23:08	LBA Status Descriptor Entry 0: This field contains the first LBA Status Descriptor Entry in the list, if present.										
39:24	LBA Status Descriptor Entry 1: This field contains the second LBA Status Descriptor Entry in the list, if present.										
...	...										
(N*16+23): (N*16+8)	LBA Status Descriptor Entry N: This field contains the N+1 LBA Status Descriptor Entry in the list, if present.										

Figure 112: LBA Status Descriptor Entry

Bytes	Description
07:00	Descriptor Starting LBA (DSLBA): This field specifies indicates the 64-bit address of the first logical block of the LBA range for which this LBA Status Descriptor Entry reports LBA status.
11:08	<p>Number of Logical Blocks (NLB): This field contains indicates the number of contiguous logical blocks reported in this LBA Status Descriptor Entry. The controller should return the largest aggregated possible value in this field. perform the action specified in the Action Type field in such a way that the value in this field reports the largest number of contiguous logical blocks possible (i.e., multiple consecutive LBA Status Descriptor Entries should not report contiguous LBAs that span those entries, but rather, LBA Status Descriptor Entries should be consolidated into the fewest number of LBA Status Descriptor Entries possible). This is a 0's based value.</p>
12	Reserved

Figure 112: LBA Status Descriptor Entry

Bytes	Description																							
13	Status: This field contains additional status information about this LBA range.																							
	<table><tr><th>Bits</th><th>Definition</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>4</td><td>If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.</td></tr><tr><td>0</td><td>If set to '1', a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.</td></tr><tr><td rowspan="5">1:0</td><td>These bits indicate information about the logical blocks indicated in this LBA Status Descriptor Entry.</td></tr><tr><td><table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Each logical block may:<ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors;be a logical block for which the most recent write to the logical block was a Write Uncorrectable command; orbe read successfully.</td></tr><tr><td>01b</td><td>Each logical block may:<ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors; orbe a logical block for which the most recent write to the logical block was a Write Uncorrectable command.</td></tr><tr><td>10b</td><td>Reserved</td></tr><tr><td>11b</td><td>Each logical block is a:<ul style="list-style-type: none">logical block for which the most recent write to the logical block was a Write Uncorrectable command.</td></tr></table></td></tr><tr><td>15:14</td><td>Reserved</td></tr></table>	Bits	Definition	7:2	Reserved	4	If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.	0	If set to '1', a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.	1:0	These bits indicate information about the logical blocks indicated in this LBA Status Descriptor Entry.	<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Each logical block may:<ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors;be a logical block for which the most recent write to the logical block was a Write Uncorrectable command; orbe read successfully.</td></tr><tr><td>01b</td><td>Each logical block may:<ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors; orbe a logical block for which the most recent write to the logical block was a Write Uncorrectable command.</td></tr><tr><td>10b</td><td>Reserved</td></tr><tr><td>11b</td><td>Each logical block is a:<ul style="list-style-type: none">logical block for which the most recent write to the logical block was a Write Uncorrectable command.</td></tr></table>	Value	Definition	00b	Each logical block may: <ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors;be a logical block for which the most recent write to the logical block was a Write Uncorrectable command; orbe read successfully.	01b	Each logical block may: <ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors; orbe a logical block for which the most recent write to the logical block was a Write Uncorrectable command.	10b	Reserved	11b	Each logical block is a: <ul style="list-style-type: none">logical block for which the most recent write to the logical block was a Write Uncorrectable command.	15:14	Reserved
	Bits	Definition																						
	7:2	Reserved																						
	4	If set to '1', this LBA range describes LBAs written with a Write Uncorrectable command. If cleared to '0', this LBA range may or may not describe LBAs written with a Write Uncorrectable command.																						
	0	If set to '1', a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status. If cleared to '0', the controller has not detected that a Copy, Read, Verify, or Compare command to each LBA reported in this LBA Status Descriptor may result in a command completion with Unrecovered Read Error status.																						
	1:0	These bits indicate information about the logical blocks indicated in this LBA Status Descriptor Entry.																						
		<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00b</td><td>Each logical block may:<ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors;be a logical block for which the most recent write to the logical block was a Write Uncorrectable command; orbe read successfully.</td></tr><tr><td>01b</td><td>Each logical block may:<ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors; orbe a logical block for which the most recent write to the logical block was a Write Uncorrectable command.</td></tr><tr><td>10b</td><td>Reserved</td></tr><tr><td>11b</td><td>Each logical block is a:<ul style="list-style-type: none">logical block for which the most recent write to the logical block was a Write Uncorrectable command.</td></tr></table>	Value	Definition	00b	Each logical block may: <ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors;be a logical block for which the most recent write to the logical block was a Write Uncorrectable command; orbe read successfully.	01b	Each logical block may: <ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors; orbe a logical block for which the most recent write to the logical block was a Write Uncorrectable command.	10b	Reserved	11b	Each logical block is a: <ul style="list-style-type: none">logical block for which the most recent write to the logical block was a Write Uncorrectable command.												
		Value	Definition																					
		00b	Each logical block may: <ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors;be a logical block for which the most recent write to the logical block was a Write Uncorrectable command; orbe read successfully.																					
		01b	Each logical block may: <ul style="list-style-type: none">report Unrecovered Read Error status as a result of media errors; orbe a logical block for which the most recent write to the logical block was a Write Uncorrectable command.																					
	10b	Reserved																						
	11b	Each logical block is a: <ul style="list-style-type: none">logical block for which the most recent write to the logical block was a Write Uncorrectable command.																						
15:14	Reserved																							

The Descriptor Starting LBA (DSLBA) field in the first LBA Status Descriptor **e**Entry returned in the LBA Status Descriptor List shall contain the lowest numbered LBA that is greater than or equal to the value specified in the Starting LBA field in the Get LBA Status command.

For subsequent LBA Status Descriptor **e**Entries, the contents of the Descriptor Starting LBA field shall contain the value of the lowest numbered LBA meeting the requirements for the specified Action Type value that is greater than the sum of the values in:

- the Descriptor Starting LBA field in the previous LBA Status Descriptor **e**Entry; and
- the Number of Logical Blocks field in the previous LBA Status Descriptor **e**Entry.

...

Modify a portion of section 5.2 as shown below:

5.2 End-to-end Data Protection

...

5.2.1.4 Storage Tag and Logical Block Reference Tag from Storage and Reference Space

The Storage Tag Size (STS) field in the Identify Namespace data structure allows the separation of the Storage and Reference Space field in the protection information ~~formats to be separated~~ into a Storage Tag field and a Logical Block Reference Tag field as shown in Figure 122.

...

5.2.2.5 Protection Information and Copy commands

...

For the portion of the Copy command that transfers data and protection information to the LBAs starting at the SDLBA field (refer to Figure 29), the protection information operations performed by the controller are controlled by the PRINFOW field in Copy command Dword 12 (refer to Figure 30) and parallels the Write command protection information checks (refer to section 5.2.2.1) as follows:

- The logical block data and metadata are transferred from the controller to the NVM.
- As the logical block data and metadata passes through the controller, the protection information is handled as described in section 5.2.2.1 ~~8.3.1.1~~.

...

5.2.3 Control of Protection Information Checking - PRCHK

...

- If the Application Tag Check bit of the PRCHK field is set to '1', then the controller compares unmasked bits in the protection information Application Tag field to the Logical Block Application Tag (LBAT) field in the command. A bit in the protection information Application Tag field is masked if the corresponding bit is cleared to '0' in the Logical Block Application Tag Mask (LBATM) field of the command or the Expected Logical Block Application Tag Mask (ELBATM) field. If a Storage Tag field is defined in the protection information (refer to section 5.2.1.4) and the Storage Tag Check bit in the command is set to '1', then the controller compares unmasked bits in the Storage Tag field to the Logical Block Storage Tag (LBST) field of the command. A bit in the Storage Tag field is masked (*i.e., not compared*) if the corresponding bit is cleared to '0' in the Storage Tag Mask (LBSTM) field in the NVM Command Set Identify Namespace data structure (refer to Figure 100).

Modify a portion of section 5.8 as shown below:

5.8 Command Set Specific Capability

...

5.8.1 Get LBA Status

Potentially Unrecoverable LBAs are LBAs that, when read, may result in the command that caused the media to be read being aborted with a *status code of* Unrecovered Read Error ~~status~~. The Get LBA Status capability provides the host with the ability to identify Potentially Unrecoverable LBAs. The logical block data is able to be recovered from another location and re-written.

To support the Get LBA Status capability, the NVM subsystem shall:

- indicate support for the Get LBA Status capability in the Optional Admin Command Support (OACS) field in the Identify Controller data structure;
- indicate support for LBA Status Information **Alert** Notices in the Optional Asynchronous Events Supported (**OAES**) field in the Identify Controller data structure;
- support the LBA Status Information log page;
- indicate support for the Log Page Offset and extended Number of Dwords (i.e., 32 bits rather than 12 bits) in the Log Page Attributes field of the Identify Controller data structure;
- support the LBA Status Information Attributes Feature;
- support the Get LBA Status command; and
- support the LBA Status Information Alert Notices event.

Prior to using the Get LBA Status capability:

- The host should use the Get Features and Set Features commands with the LBA Status Information Attributes Feature (refer to section 4.1.3.3) to retrieve and optionally configure the LBA Status Information Report Interval; and
- If the host wishes to receive LBA Status Information Alert asynchronous events, the host should enable LBA Status Information **Alert** Notices (refer to Figure 88).

If LBA Status Information **Alert** Notices are enabled, the controller shall send an LBA Status Information Alert asynchronous event if:

- there are Tracked LBAs and:
 - the LBA Status Information Report Interval condition has been exceeded; or
 - an implementation specific aggregate threshold, if any exists, of Tracked LBAs has been exceeded;
- or
- a component (e.g., die or channel) failure has occurred that may result in the controller aborting commands with a **status code** of Unrecovered Read Error **status**.

...

The LBA Status Information Report Interval is restarted by the controller when the host issues a Get Log Page **command** for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0'. ~~Reading the Issuing a~~ Get Log Page **command** for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0' causes an outstanding LBA Status Information **Alert** asynchronous event to be cleared if there is one outstanding on the controller.

When the host re-reads the header of the LBA Status Information log page with the Retain Asynchronous Event bit cleared to '0', the host should ensure that the LBA Status Generation Counter matches the original value read. If these values do not match, there is newer LBA Status **Information** ~~Log Page~~ data available than ~~that which was the data returned to the host the last previous~~ time the host read the LBA Status Information log **page**. In this case, the host is not required to wait for the LBA Status Information Poll Interval (LSIPI) to pass before re-reading the LBA Status Information log page.

The host decides when to send Get LBA Status commands and when to recover the LBAs identified by the Get LBA Status commands, relative to when the host ~~reads the~~ **issues a** Get Log Page **command** for Log Identifier 0Eh with the Retain Asynchronous Event bit cleared to '0'. Section 5.8.1.1 describes some example host implementations.

...

5.8.1.1 Sample Get LBA Status Host Software Implementations (Informative)

5.8.1.1.1 Example Flow #1

- 1) Read ~~all~~ the LBA Status Information log page ~~data~~ with RAE bit set to '1';
- 2) Complete all necessary Get LBA Status commands;
- 3) Complete all necessary recovery of the affected user data by rewriting that data; and
- 4) Read the LBA Status Information log page ~~data~~ header with RAE bit cleared to '0'.

5.8.1.1.2 Example Flow #2

- 1) Read ~~all~~ the LBA Status Information log page ~~data~~ with RAE bit set to '1';
- 2) Read the LBA Status Information log page ~~data~~ with RAE bit cleared to '0';
- 3) Issue some host-determined subset of the Get LBA Status commands indicated by the log page data;
- 4) ...

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

...

2.1.1.2.1.2 Reading in Sequential Write Required Zones

...

If the Read Across Zone Boundaries bit is set to '1' in the Zoned Namespace Command Set specific Identify Namespace data structure (refer to Figure 48), then User Data Read Access Commands are allowed to specify an LBA range containing logical blocks in more than one zone.

...

2.1.1.3.2 ZSIO:Implicitly Opened state

...

Transition ZSIO:ZSF: The zone shall transition from the ZSIO:Implicitly Opened state to the ZSF:Full state:

- a) as a result of successful completion of a Zone Management Send command with a Zone Send Action of Finish Zone;
- b) as a result of ~~successful completion of~~ a write operation that writes one or more logical blocks that causes the zone to reach its writeable zone capacity;
- c) due to a Zone Active Excursion (refer to section 5.6); and
- d) as a result of the zoned namespace becoming ~~es~~ing write protected (refer to the Namespace Write Protection section in the NVMe Base Specification).

2.1.1.3.3 ZSEO:Explicitly Opened state

...

Transition ZSEO:ZSF: The zone shall transition from the ZSEO:Explicitly Opened state to the ZSF:Full state:

- a) as a result of successful completion of a Zone Management Send command with a Zone Send Action of Finish Zone;
- b) as a result of ~~successful completion of~~ a write operation that writes one or more logical blocks that causes the zone to reach its writeable zone capacity;
- c) due to a Zone Active Excursion (refer to section 5.6); or
- d) as a result of the zoned namespace becoming ~~es~~ing write protected (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 Zoned Namespace Command Set

...

4.1.1 Asynchronous Event Request Command

...

Figure 42: Asynchronous Event Information – Notice, Zoned Namespace Command Set

Value	Description
EFh	<p>Zone Descriptor Changed: The Zone Descriptor data structure for a zone changed in a specific zoned namespace (refer to Figure 43) has changed. The Zone Descriptor of the zone that changed is indicated in the Changed Zone List log page. To clear this event, host software reads the Zone Changed List log page using the Get Log Page command with the Retain Asynchronous Event bit cleared to '0'.</p> <p>For a specific zone, a Zone Descriptor data structure change caused by that occurred for any of the following reasons, shall not generate a Zone Descriptor Changed event and shall not cause modifications to the Changed Zones List log page:</p> <ul style="list-style-type: none">a) a Zone Management Send command that specified that zone;b) a Zone Management Send command that specified all zones;c) a write operation that transitioned that zone:<ul style="list-style-type: none">i. from the ZSE:Empty state to the ZSIO:Implicitly Opened state;ii. from the ZSIO:Implicitly Opened state to the ZSF:Full state;iii. from the ZSEO:Explicitly Opened state to the ZSF:Full state; and oriv. from the ZSC:Closed state to the ZSIO:Implicitly Opened state;d) the controller transitioning that zone to the ZSF:Full state due to an NVM Subsystem Reset; ande) the controller transitioning that zone to the ZSC:Closed state (refer to section 2.1.1.4.1).

...

4.1.1.1 Command Completion

A completion queue entry (CQE) is posted to the Admin Completion Queue if there is an asynchronous event to report to the host. **A controller and** implements the same logic defined for the Asynchronous Event Request command (~~refer to in the NVMe Base Specification~~), with the following addition:

Figure 434: Asynchronous Event Request – Completion Queue Entry Dword 1

Bytes	Description
3:0	<p>Namespace Identifier (NSID): If the Asynchronous Event Request command completed successfully returned a Zoned Namespace Command Set Notice event (refer to Figure 42), then this field indicates the namespace identifier that the asynchronous event occurred on. If the Asynchronous Event Request command returned an event other than a Zoned Namespace Command Set Notice event, then this field is reserved.</p> <p>If the Asynchronous Event Request command did not complete successfully, then this field is reserved.</p>

...

4.1.5.2 I/O Command Set Specific Identify Controller **d**Data **s**Structure (CNS 06h, CSI 02h)

...

Modify a portion of Annex section A.4 as shown below:

A.4 ~~Capacity and Sizes~~Partial Failures

...

A.4.1 Overview

The zone state ZSRO:Read Only provides the ability for a host to continue using a zoned namespace after part of the capacity of the zone has stopped operating (e.g., the controller transitions a zone to the ZSRO:Read Only state as a response to a media failure).

After a zone enters the ZSRO:Read Only state, the host should perform the following actions:

- 1) Transfer the data from that zone to another location (e.g., by using the Copy command); and
- 2) Transition that zone to the ZSO:Offline state by issuing a Zone Management Send command with a Zone Send Action of Offline Zone.

~~The host may explicitly transition a zone in the ZSRO:Read Only state to the ZSO:Offline state by issuing a Zone Management Send command with Offline Zone Send Action specified.~~

...

Description of NVM Express Key Value Command Set specification changes

Modify a portion of section 2.1 as shown below:

2.1 Theory of operation

...

2.1.1 Namespaces

...

The ~~following relationship holds: Namespace Size >=~~ Namespace Utilization (NUSE) field shall be less than or equal to the Namespace Size (NSZE) field.

...

Modify a portion of section 4.1 as shown below:

4.1 Admin Command behavior for the Key Value Command Set

...

4.1.5.1 I/O Command Set sSpecific Identify Namespace dData sStructure (CNS 05h, CSI 01h)

...

Figure 36: Identify – I/O Command Set sSpecific Identify Namespace Data Structure, Key Value Type Specific

Bytes	O/M ¹	Description
...		
23:16	M	Namespace Utilization (NUSE): This field indicates the current number of bytes allocated in the namespace. This is the space to store KV keys and KV values. This field is smaller less than or equal to the Namespace Size field . A key value pair is allocated when it is written with a Store command. A key value pair is deallocated using the Delete command. If the controller supports Asymmetric Namespace Access Reporting (refer to the CMIC field), and the relationship between the controller and the namespace is in the ANA Inaccessible state (refer to the NVMe Base Specification) or the ANA Persistent Loss state (refer to the NVMe Base Specification), then this field shall be cleared to 0h.
...		

Figure 36: Identify – I/O Command Set ~~s~~Specific Identify Namespace Data Structure, Key Value Type Specific

Bytes	O/M ¹	Description
25	M	<p>Number of KV Formats (NKVF): This field defines the number of KV format descriptors supported by the namespace. KV formats shall be allocated in order (starting with 0) and packed sequentially. This is a 0's based value.</p> <p>The maximum number of KV formats that may be indicated as supported is 16. The supported KV formats are indicated in bytes 72 to 327 in this data structure.</p> <p>The KV Format fields with a a Format i index beyond the value set in this field are invalid and not supported. KV Formats that are valid, but not currently available may be indicated by setting clearing the KV Key Max Length field to 0h and clearing the KV Value Max Length field both to 0000h for that KV Format.</p>
26	O	<p>Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC): Refer to the NMIC field in the I/O Command Set Independent Identify Namespace data structure in the NVMe Base Specification.</p>
...		

...

4.1.5.2 ~~Identify~~ I/O Command Set ~~s~~Specific **Identify** Controller ~~d~~Data ~~s~~Structure (CNS 06h, CSI 01h)

...

Description of NVM Express PCI Express Transport specification changes

Modify a portion of 3.5 as shown below:

3.5 Interrupts

The interrupt architecture allows for efficient reporting of interrupts such that the host may service interrupts through the least amount of overhead.

...

3.5.1 Pin Based, Single MSI, and Multiple MSI Behavior

This is the mode of interrupt operation if any of the following conditions are met:

- Pin-based interrupts are being used – MSI (i.e., MSICAP.MC.MSIE='0') and MSI-X are disabled;
- Single MSI is being used – MSI is enabled (i.e., MSICAP.MC.MSIE='1'), MSICAP.MC.MME=000b, and MSI-X is disabled; or
- Multiple MSI is being used – multiple message MSI is enabled (i.e., MSICAP.MC.MSIE='1'), MSICAP.MC.MME is set to a value between 001b and 101b inclusive, and MSI-X is disabled.

Within the controller there is an interrupt status register (IS) that is not visible to the host. In this mode, the IS register determines whether the PCI interrupt line shall be driven active or an MSI message shall be sent. Each bit in the IS register corresponds to an interrupt vector. The IS bit is set to '1' when the following conditions are true:

- There is one or more unacknowledged Completion Queue entries in a Completion Queue that utilizes this interrupt vector;
- The Completion Queue(s) with unacknowledged Completion Queue entries has interrupts enabled in the Create I/O Completion Queue command; and
- The corresponding INTM bit exposed to the host is cleared to '0', indicating that the interrupt is not masked.

For single and multiple MSI, the INTM register masks interrupt delivery prior to MSI logic. As such, an interrupt on a vector masked by INTM does not cause the corresponding Pending bit to assert within the MSI Capability Structure.

If MSIs are not enabled, IS[0] being set to '1' causes the PCI interrupt line to be active (electrical '0'). If MSIs are enabled, any change to the IS register that causes an unmasked status bit to transition from '0' to '1' or clearing of a mask bit to '0' whose corresponding status bit is set to '1' shall cause an MSI to be sent. Therefore, while in wire mode, a single wire remains active, while in MSI mode, several messages may be sent, as each edge triggered event on a port shall cause a new message.

If pin-based interrupts or single message MSI interrupts are used, host software interrogates the Completion Queue(s) for new completion queue entries. If multiple message MSI is in use, then the interrupt vector indicates the Completion Queue(s) with possible new command completions for the host to process.

~~In order to~~ To clear an interrupt for a particular interrupt vector, host software acknowledges all Completion Queue entries for Completion Queues associated with the interrupt vector by updating the CQ Head doorbell register (refer to the Memory-Based Transport Model section in the NVMe Base Specification).

...

3.5.1.1 Differences Between Pin Based and MSI Interrupts

Single MSI is similar to the pin-based interrupt behavior mode. The primary difference is the method of reporting the interrupt. Instead of communicating the interrupt through an INTx virtual wire, an MSI message is generated to the host. Unlike INTx virtual wire interrupts which are level sensitive, MSI interrupts are edge sensitive.

Pin-based and single MSI only use one interrupt vector. Multiple MSI may use up to 32 interrupt vectors.

For multiple MSI, the controller advertises the requested number of MSI interrupt vectors in the Multiple Message Capable (MMC) field in the Message Signaled Interrupt Message Control (MC) register. The MSICAP.MC.MMC field represents a power-of-2 wrapper on the number of requested vectors. For example, if three vectors are requested, then the MSICAP.MC.MMC field shall be '010' (i.e., four vectors).

Multiple-message MSI allows completions to be aggregated on a per vector basis. If sufficient MSI vectors are allocated, each Completion Queue(s) may send its own interrupt message, as opposed to a single message for all completions.

3.5.2 MSI-X Based Behavior

MSI-X is the preferred interrupt behavior to use. The following configuration describes this mode of interrupt operation:

- Multiple-message MSI is disabled (i.e., MSICAP.MC.MSIE is cleared to '0') and MSICAP.MC.MME is cleared to 000b; and
- MSI-X is enabled.

MSI-X, similar to multiple-message MSI, allows completions to be aggregated on a per vector basis. However, the maximum number of vectors is 2KiB. MSI-X also allows each interrupt to send a unique message data corresponding to the vector.

MSI-X allows completions to be aggregated on a per vector basis. Each Completion Queue(s) may send its own interrupt message, as opposed to a single message for all completions.

When generating an MSI-X message, the following checks occur before generating the message:

- The Function Mask bit in the MSI-X Message Control register is cleared to '0'; and
- The corresponding vector mask in the MSI-X table structure is cleared to '0'.

If either of the mask bits are set to '1', the corresponding pending bit in the MSI-X PBA structure is set to '1' to indicate that an interrupt is pending for that vector. The MSI for that vector is later generated when both the mask bits are cleared to '0'.

It is recommended that the interrupt vector associated with the CQ(s) being processed be masked during processing of Completion Queue entries within the CQ(s) to avoid spurious and/or lost interrupts. The interrupt mask table defined as part of MSI-X should be used to mask interrupts.

The interrupt vector indicates the Completion Queue(s) with possible new command completions for the host to process.

...

Description of NVM Express TCP Transport specification changes

Modify a portion of section 3.3 as shown below:

3.3 Data Transfer Model

...

3.3.1.1 PDU Header and Data Digests

NVMe/TCP facilitates an optional PDU Header Digest (HDGST) and Data Digest (DDGST). The presence of each digest is negotiated at the connection establishment.

The host requests the use of a header digest by setting the HDGST_ENABLE flag in the ICRReq PDU. The controller may accept ~~(or reject)~~ the use of a header digest by setting ~~to '1' (or clearing to '0')~~ the HDGST_ENABLE flag ~~to '1'~~ in the ICRResp PDU. ~~The controller may reject the use of a header digest by clearing the HDGST_ENABLE flag to '0' in the ICRResp PDU.~~ The PDU Header Digest is enabled if HDGST_ENABLE flag is set ~~to '1'~~ in both the ICRReq PDU and ~~the~~ ICRResp PDUs. If the PDU Header Digest is enabled, then all the subsequent PDUs transferred in this connection except a H2CTermReq PDU and a C2HTermReq PDUs shall contain a valid HDGST field and ~~have~~ the HDGSTF flag, ~~if defined, shall be~~ set to '1' in the PDU Header FLAGS field. If the PDU Header Digest is enabled, the header digest is contained within the HDGST field of the PDU and protects the PDU Header. If PDU Header Digest is not enabled, then all subsequent PDUs shall not contain a HDGST field, and ~~shall have~~ the HDGSTF flag, if defined, ~~shall be~~ cleared to '0' in the PDU Header FLAGS field.

~~If the PDU Header Digest is enabled and a subsequent PDU transferred in this connection other than a H2CTermReq PDU or a C2HTermReq PDU has the HDGSTF flag, if defined, cleared to '0', then the receiver shall treat that PDU as if a fatal transport error (refer to section 3.5.1) has occurred. If the PDU Header Digest is disabled and a subsequent PDU is received with the HDGSTF flag, if defined, set to '1', then the receiver shall treat that PDU as if a fatal transport error has occurred.~~

The host requests the use of a data digest by setting the DDGST_ENABLE flag in the ICRReq PDU. The controller may accept ~~(or reject)~~ the use of a data digest by setting ~~to '1' (or clearing to '0')~~ the DDGST_ENABLE flag ~~to '1'~~ in the ICRResp PDU. ~~The controller may reject the use of a data digest by clearing the DDGST_ENABLE flag to '0' in the ICRResp PDU.~~ The PDU Data Digest is enabled if the DDGST_ENABLE flag is set ~~to '1'~~ in both the ICRReq PDU and ~~the~~ ICRResp PDUs. If the PDU Data Digest is enabled, then Command Capsule PDUs containing in-capsule data, H2CData PDUs, and C2HData PDUs transferred in this connection shall contain a valid DDGST field and ~~have~~ the DDGSTF flag, ~~if defined, shall be~~ set to '1' in the PDU Header FLAGS field. If the PDU Data Digest is not enabled, then these PDUs shall not contain a DDGST field and ~~shall have~~ the DDGSTF flag, ~~if defined, shall be~~ cleared to '0' in the PDU Header FLAGS field. If data digest is enabled, the data digest is contained within the DDGST field of the PDU and protects the PDU Data.

~~If the PDU Data Digest is enabled and a Command Capsule PDU containing in-capsule data, a H2CData PDU, or a C2HData PDU transferred in this connection has the DDGSTF flag, if defined, cleared to '0', then the receiver shall treat that PDU as if a fatal transport error (refer to section 3.5.1) has occurred. If the PDU Data Digest is disabled and any of these PDUs are received with the DDGSTF flag, if defined, set to '1', then the receiver shall treat that PDU as if a fatal transport error has occurred.~~

If a host requests the use of header digest or data digest in the ICRReq PDU, but the use of the digest was not enabled by the controller in the ICRResp PDU, then the host may refuse the connection establishment and terminate the NVMe/TCP connection (refer to section 3.1).

If a host did not request the use of header digest or data digest in the ICRReq PDU but the use of the digest was enabled by the controller in the ICRResp PDU, then the host shall treat that ICRResp PDU as a fatal

transport error (refer to section 3.5) with the Fatal Error Status field set to “Invalid PDU Header Field” and the Additional Error Information field containing the HDGST or DDGST field byte offset.

...

Modify a portion of section 3.6 as shown below:

3.6 Transport Specific Content

...

3.6.2.6 Command Capsule PDU (CapsuleCmd)

Figure 24: Command Capsule PDU (CapsuleCmd)

Bytes		PDU Section	Description								
00		CH	PDU-Type: 04h								
01			FLAGS:								
			<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1</td><td>DDGSTF: If set to '1', then a-valid the DDGST value field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present.</td></tr><tr><td>0</td><td>HDGSTF: If set to '1', then a-valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.</td></tr></table>	Bits	Description	7:2	Reserved	1	DDGSTF: If set to '1', then a-valid the DDGST value field follows the PDU Data and contains a valid value . If cleared to '0', then the DDGST field is not present.	0	HDGSTF: If set to '1', then a-valid the HDGST value field follows the PDU Header and contains a valid value . If cleared to '0', then the HDGST field is not present.
			Bits	Description							
			7:2	Reserved							
1	DDGSTF: If set to '1', then a-valid the DDGST value field follows the PDU Data and contains a valid value . If cleared to '0', then the DDGST field is not present.										
0	HDGSTF: If set to '1', then a-valid the HDGST value field follows the PDU Header and contains a valid value . If cleared to '0', then the HDGST field is not present.										
02		HLEN: Fixed length of 72 bytes (i.e., 48h).									
03		PDO: Data offset within PDU (i.e., the offset from byte 0 to the CCICD field; the value of 'N'). This value shall be a multiple of the data alignment specified by the CPDA field in the ICRsp PDU (refer to section 3.6.2.3) that was previously sent by the controller on this TCP connection.									
07:04		PLEN: Total length of PDU (including PDU-Header CH, PSH, HDGST, PAD, DATA, and DDGST) in bytes.									
71:08		PSH	NVMe-oF Command Capsule SQE (CCSQE): Command Capsule SQE.								
HDGSTF=1	HDGSTF=0										
75:72	Not present	HDGST	HDGST: If the HDGSTF bit is set to '1' in the FLAGS field, this field is present and contains the a valid header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.								
N - 1:76	N - 1:72	PAD	PAD: If in-capsule data is present, the length of this field shall be the necessary number of bytes required to achieve the alignment specified by the CPDA field (refer to section 3.6.2.3).								
M - 1:N		DATA	NVMe-oF In-Capsule Data (CCICD): This field contains the in-capsule data, if any, of the NVMe-oF Command Capsule.								
DDGSTF=1	DDGSTF=0										
M + 3:M	Not present	DDGST	Data Digest (DDGST): If the DDGSTF bit is set to '1' in the FLAGS field, and the CCICD field is present, then this field contains the data digest (refer to section 3.3.1.1) of the CCICD field (i.e., the in-capsule data). If the DDGSTF bit is cleared to '0', then this field is not present.								

3.6.2.7 Response Capsule PDU (CapsuleResp)

Figure 25: Response Capsule PDU (CapsuleResp)

Bytes		PDU Section	Description						
00		CH	PDU-Type: 05h						
01			FLAGS:						
			<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:1</td><td>Reserved</td></tr><tr><td>0</td><td>HDGSTF: If set to '1', then a valid HDGST value follows the PDU Header. If cleared to '0', then the HDGST field is not present.</td></tr></table>	Bits	Description	7:1	Reserved	0	HDGSTF: If set to '1', then a valid HDGST value follows the PDU Header. If cleared to '0', then the HDGST field is not present.
			Bits	Description					
			7:1	Reserved					
0	HDGSTF: If set to '1', then a valid HDGST value follows the PDU Header. If cleared to '0', then the HDGST field is not present.								
HLEN: Fixed length of 24 bytes (i.e., 18h).									
02		PDO: Reserved							
03		PLEN: Length of PDU Header CH, PSH, and HDGST, (if present), in bytes.							
07:04									
23:08			NVMe-oF Response Capsule CQE (RCCQE): Response Capsule CQE.						
HDGSTF=1	HDGSTF=0								
27:24	Not present	HDGST	HDGST: If the HDGSTF bit is set to '1' in the FLAGS field, this field is present and contains the a valid header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.						

3.6.2.8 Host To Controller Data Transfer PDU (H2CData)

Figure 26: Host To Controller Data Transfer PDU (H2CData)

Bytes		PDU Section	Description										
00		CH	PDU-Type: 06h										
01			FLAGS:										
			<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:3</td><td>Reserved</td></tr><tr><td>2</td><td>LAST_PDU: If set to '1', indicates the PDU is the last in the set of H2CData PDUs that correspond to the same R2T PDU.</td></tr><tr><td>1</td><td>DDGSTF: If set to '1', then a valid the DDGST value field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present.</td></tr><tr><td>0</td><td>HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.</td></tr></table>	Bits	Description	7:3	Reserved	2	LAST_PDU: If set to '1', indicates the PDU is the last in the set of H2CData PDUs that correspond to the same R2T PDU.	1	DDGSTF: If set to '1', then a valid the DDGST value field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present.	0	HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.
			Bits	Description									
			7:3	Reserved									
			2	LAST_PDU: If set to '1', indicates the PDU is the last in the set of H2CData PDUs that correspond to the same R2T PDU.									
1	DDGSTF: If set to '1', then a valid the DDGST value field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present.												
0	HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.												
02			HLEN: Fixed length of 24 bytes (i.e., 18h).										
03			PDO: Data Offset within PDU (i.e., the offset from byte 0 to the PDU-Data field; the value of 'N'). This value shall be a multiple of the data alignment specified by the CPDA field in the ICResp PDU (refer to section 3.4.2 3.6.2.3) that was previously sent by the controller on this TCP connection.										
07:04		PLEN: Total length of PDU (including PDU-Header CH, PSH, HDGST, PAD, DATA, and DDGST) in bytes.											
09:08		PSH	Command Capsule CID (CCCID): This field contains the SQE.CID value of the Command Capsule PDU associated with the Command Data Buffer.										
11:10			Transfer Tag (TTAG): This field contains the Transfer Tag of the corresponding R2T received by the host.										
15:12			Data Offset (DATAO): Byte offset from start of Command Data Buffer to the first byte to transfer. This value shall be a multiple of dwords.										
19:16			Data Length (DATAL): PDU-DATA PDU-Data field length in bytes (i.e., the value of M-N). This value shall be a multiple of dwords.										
23:20			Reserved										
HDGSTF=1	HDGSTF=0												
27:24	Not present	HDGST	HDGST: If the HDGSTF bit is set to '1' in the FLAGS field, this field is present and contains the a valid header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.										
N - 1: N 28	N - 1:24	PAD	PAD: The length of this field shall be the necessary number of bytes required to achieve the alignment specified by the CPDA field (refer to section 3.6.2.3).										
M - 1:N		DATA	PDU-Data: This field contains the contents of the Command Data Buffer being transferred. The length of this field is a multiple of dwords.										
DDGSTF=1	DDGSTF=0												
M + 3:M	Not present	DDGST	Data Digest (DDGST): If the DDGSTF bit is set to '1' in the FLAGS field, this field is present and contains the a valid data digest (refer to section 3.3.1.1). If the DDGSTF bit is cleared to '0', then this field is not present.										

3.6.2.9 Controller To Host Data Transfer PDU (C2HData)

Figure 27: Controller To Host Data Transfer PDU (C2HData)

Bytes		PDU Section	Description												
00		CH	PDU-Type: 07h												
01			FLAGS:												
			<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:4</td><td>Reserved</td></tr><tr><td>3</td><td>SUCCESS: If set to '1', indicates that the command referenced by CCCID was completed successfully with no other information and that no Response Capsule PDU is sent by the Controller.</td></tr><tr><td>2</td><td>LAST_PDU: If set to '1', indicates the PDU is the last C2HData PDU sent in response to a Command Capsule PDU.</td></tr><tr><td>1</td><td>DDGSTF: If set to '1', then a valid the DDGST value field follows the PDU Data and contains a valid value. If cleared to '0', then the DDGST field is not present.</td></tr><tr><td>0</td><td>HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.</td></tr></table>	Bits	Description	7:4	Reserved	3	SUCCESS: If set to '1', indicates that the command referenced by CCCID was completed successfully with no other information and that no Response Capsule PDU is sent by the Controller.	2	LAST_PDU: If set to '1', indicates the PDU is the last C2HData PDU sent in response to a Command Capsule PDU.	1	DDGSTF: If set to '1', then a valid the DDGST value field follows the PDU Data and contains a valid value . If cleared to '0', then the DDGST field is not present.	0	HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value . If cleared to '0', then the HDGST field is not present.
			Bits	Description											
			7:4	Reserved											
			3	SUCCESS: If set to '1', indicates that the command referenced by CCCID was completed successfully with no other information and that no Response Capsule PDU is sent by the Controller.											
			2	LAST_PDU: If set to '1', indicates the PDU is the last C2HData PDU sent in response to a Command Capsule PDU.											
1	DDGSTF: If set to '1', then a valid the DDGST value field follows the PDU Data and contains a valid value . If cleared to '0', then the DDGST field is not present.														
0	HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value . If cleared to '0', then the HDGST field is not present.														
02			HLEN: Fixed length of 24 bytes (i.e., 18h).												
03		PDO: Data offset within PDU (i.e., the offset from byte 0 to the PDU-Data field; the value of 'N'). This value shall be a multiple of the data alignment specified by the HPDA field in the ICReq PDU (refer to section 3.6.1.1 3.6.2.2) that was previously sent by the host on this TCP connection.													
07:04		PLEN: Total length of PDU (i.e., including PDU-Header CH, PSH, HDGST, PAD, DATA, and DDGST) in bytes.													
09:08		PSH	Command Capsule CID (CCCID): This field contains the SQE.CID value of the Command Capsule PDU associated with the host-resident data.												
11:10			Reserved												
15:12			Data Offset (DATAO): Byte offset from start of host-resident data to the first byte to transfer. This value shall be dword aligned.												
19:16			Data Length (DATAL): PDU-DATA -PDU-Data field length in bytes (i.e., the value of M-N). This value shall be dword aligned.												
23:20			Reserved												
HDGSTF=1	HDGSTF=0														
27:24	Not present	HDGST	HDGST: If the HDGSTF bit is set to '1' in the FLAGS field, this field is present and contains the a valid header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.												
N - 1: N 28	N - 1:24	PAD	PAD: If the HPDA field (refer to section 3.6.2.2) is set to a non-zero value, then this field is padded as specified by HPDA. the length of this field shall be the necessary number of bytes required to achieve the alignment specified by the HPDA field.												
M - 1:N		DATA	PDU-Data: This field contains the host-resident data being transferred. The length of this field is a multiple of dwords.												
DDGSTF=1	DDGSTF=0														
M + 3:M	Not present	DDGST	Data Digest (DDGST): If the DDGSTF bit is set to '1' in the FLAGS field, this field is present and contains a valid d Data digest (refer to section 3.3.1.1) of the PDU-Data field. If the DDGSTF bit is cleared to '0', then this field is not present.												

3.6.2.10 Ready to Transfer PDU (R2T)

Figure 28: Ready to Transfer PDU (R2T)

Bytes	PDU Section	Description						
00	CH	PDU-Type: 09h						
01		FLAGS:						
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:1</td><td>Reserved</td></tr><tr><td>0</td><td>HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.</td></tr></table>	Bits	Description	7:1	Reserved	0	HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.
		Bits	Description					
7:1		Reserved						
0		HDGSTF: If set to '1', then a valid the HDGST value field follows the PDU Header and contains a valid value. If cleared to '0', then the HDGST field is not present.						
02	HLEN: Fixed length of 24 bytes (i.e., 18h).							
03	PDO: Reserved							
07:04	PLEN: Length of PDU Header CH, PSH, and HDGST, (if present) , in bytes.							
09:08	PSH	Command Capsule CID (CCCID): This field contains the SQE.CID value of the Command Capsule PDU associated with the host-resident data.						
11:10		Transfer Tag (TTAG): This field contains a controller generated tag. The rules of the tag generation are completely up to the controller's discretion outside the scope of this specification.						
15:12		Requested Data Offset (R2TO): Byte offset from the start of the host-resident data to the first byte to transfer. This value shall be dword aligned.						
19:16		Requested Data Length (R2TL): Number of bytes of Command Data Buffer requested by the controller. This value shall be dword aligned.						
23:20		Reserved						
HDGSTF=1		HDGSTF=0						
27:24	Not present	HDGST	HDGST: If the HDGSTF bit is set to '1' in the FLAGS field, this field is present and contains the a valid header digest (refer to section 3.3.1.1). If the HDGSTF bit is cleared to '0', then this field is not present.					