



#### **LEGAL NOTICE:**

© **Copyright 2007 - 2021 NVM Express, Inc. ALL RIGHTS RESERVED.**

This NVM Express revision 1.4 technical proposal 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 NVM Express revision 1.4 technical proposal 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 - 2021 NVM Express, Inc. ALL RIGHTS RESERVED." When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

#### **LEGAL DISCLAIMER:**

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

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

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

*Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.*

**NVM Express Technical Proposal for New Feature**

<b>Technical Proposal ID</b>	<b>4053a – Zoned Namespaces</b>
<b>Change Date</b>	<b>2021-04-06</b>
<b>Builds on Specification</b>	<b>NVM Express 1.4a</b>
<b>Refers to Ratified Technical Proposals</b>	<b>TP 4053: Zoned Namespaces</b> <b>TP 4040: Non-Data Transfer (non-MTDS) Command Size Limits</b> <b>TP 4056b: Namespace Types</b> <b>TP 4065a: Simple Copy Command</b> <b>TP 4105a: I/O Command Set</b> <b>Independent data structure</b>

**Technical Proposal Author(s)**

<b>Name</b>	<b>Company</b>
Christoph Hellwig, Matias Bjørling, Yoni Shternhell, Dave Landsman, Dan Helmick, Hans Holmberg, Niklas Cassel	Western Digital
Jonathan Hughes, Nick Adams, Shirish Bahirat, Michael Allison, James Harris	Intel
Laura Caulfield, Lee Prewitt, Nathan Obr, Scott Lee	Microsoft
Feng Zhu, Fei Liu, Alan Wu, Shu Li	Alibaba
Curtis Stevens, Steve Wells, Mark Carlson, John Geldman, Paul Suhler	Kioxia
James Hatfield, Curtis Stevens	Seagate
Peter Onufryk	Microchip
Fred Knight	NetApp
Paul Suhler, John Maroney, Walt Hubis	Micron
Neil Wanamaker	SK hynix
Bill Martin, Javier Gonzalez, Judy Brock	Samsung
Martin Petersen	Oracle

This proposal defines Zoned Namespaces and the associated Zoned Namespace Command Set.

## Revision History

Revision Date	Change Description
2020-06-18	<ul style="list-style-type: none"> <li>Fix byte offset for the OZCS field in the Zoned Namespace Command Set specific Identify Namespace data structure.</li> <li>Align section 5.15 (Base) to the appropriate title. Change “Identity” to “Identify”.</li> </ul>
2020-06-19	<ul style="list-style-type: none"> <li>Figure 50, modified the Formatted LBA Size (FLBAS) field from byte 25 to byte 26.</li> </ul>
2020-06-25	<ul style="list-style-type: none"> <li>Figure 50, moved the bits description into the description column.</li> </ul>
2020-07-02	<ul style="list-style-type: none"> <li>Figure 50, removed note and added NVM Command Set and ZNS Command Set into the figure.</li> </ul>
2020-07-30	<p>Proposed changes that have not yet been discussed in a ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>Clarified that Zone Append can be used to fill the zone in response to a Finish Zone Recommended.</li> <li>Removed reference to Zoned Namespace Command Set from changes to the Base spec.</li> <li>Added statement that Directives are not supported.</li> <li>Added change to TP 4056 for the Identify Namespace data structure for an Allocated Namespace ID (CNS 11h). (From Refactoring TG meeting 2020-07-29)</li> <li>Added other questions.</li> </ul>
2020-08-03	<p>Added change to TP 4056a, per Refactoring TG meeting of 2020-07-29.</p> <p>Support for Directives:</p> <ul style="list-style-type: none"> <li>Added DTYPE field to CDW12 of Zone Append command;</li> <li>Added CDW13 (with DSPEC field) to Zone Append command.</li> <li>Modified proposed section 4.8 to state that support for Directives is optional.</li> </ul> <p>Added Dataset Management (DSM) field to Zone Append command.</p> <p>Comments flagging sections to be changed by TPs 4055b and 4068.</p>
2020-08-04	<p>Changes from 2020-08-04 ZNS TG Meeting:</p> <ul style="list-style-type: none"> <li>Directives: Confirmed addition, but don't need a separate section (4.8).</li> <li>Dataset Management: Removed. This will be reconsidered in the future.</li> <li>Added section for the Changed Namespace List log page, to define which Identify Namespace data structures are covered by this log page.</li> </ul> <p>Minor editorial changes not discussed in the meeting. See comments.</p>
2020-08-10	<p>Fix typos and grammar in the following sections: 1.5.6, 1.5.7, 2.4.4, 2.5, 3.1.1, 4.4.1.2, and 4.5.</p>
2020-08-11	<p>Changes from 2020-08-11 ZNS TG Meeting:</p> <ul style="list-style-type: none"> <li>Incorporate task group feedback.</li> </ul> <p>Add “associated” to the first paragraph of Section 2.5.</p>
2020-08-18	<p>Incorporate feedback from the 2020-08-18 ZNS TG Meeting.</p>
2020-08-28	<p>Incorporate feedback from Paul</p> <p>Apply reviewed changes to the document.</p>
2020-09-04	<p>Incorporate feedback from Paul</p> <p>Improve readability wrt to Select All bit in the Zone Management Send command.</p>
2020-09-30	<p>Changes from 2020-09-29 ZNS TG Meeting:</p> <ul style="list-style-type: none"> <li>Fixed organization and naming of Identify data structures sections.</li> <li>Reorganized paragraphs in three zone action descriptions.</li> <li>Added shared namespace statement to cover resources in section 2.5.</li> <li>Moved shared namespace statements in section 5.2 and 5.3 to section 4.3 by using a shared namespace statement that applies to all Zone Management Send commands.</li> </ul>
2020-10-06	<p>Changes from 2020-10-06 ZNS TG Meeting:</p> <ul style="list-style-type: none"> <li>2.3.1.2: Updated proposed replacement text.</li> <li>4.3.1.3: Updated proposed text to that suggested by Fred.</li> <li>Fixed two incorrect cross references.</li> <li>Marked all accepted comments as done.</li> </ul>
2020-10-13	<p>Changes from 2020-10-13 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>1.5: Commented Read Operation and Write Operation for possible de-capitalization.</li> <li>2.3.1.2: Updated proposed replacement text again.</li> <li>Commented MAR and MOR fields as needing clarifying text.</li> </ul>

2020-10-20	<p>Changes from 2020-10-20 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• Merged changes from TP 4068 (pending ratification).</li> <li>• 1.5: Modified Active Zone and Open Zone definitions.</li> <li>• Fig. 8: Modified MAR and MOR fields.</li> <li>• 2.3.1.2: Final wording for read operations..</li> <li>• Figure 52: Replaced sentence in text block. More rewording needed.</li> <li>• Deleted comments which had been resolved in earlier revisions.</li> <li>• Added comments for offline discussions that are needed.</li> </ul>
2020-10-23	<p>Changes from reflector discussions:</p> <ul style="list-style-type: none"> <li>• Added TP 4068 to referenced TP list.</li> <li>• Change definitions to lower case, when not the name of a field or structure.</li> <li>• Capitalized Zone Descriptor and Zone Descriptor Extension in the text, where needed.</li> <li>• Changed “associated to” to “associated with”.</li> <li>• 4.3.1 (Zone Send Actions): Appended “in Command Dword 13” to the first instance of “Select All bit” in each subsection of 4.3.1.</li> <li>• 4.3.1.4 (Reset Zone): Fixed verbs for the Zone Descriptor.</li> <li>• 4.4 (Zone Management Receive command): Added introductory text about use of the command.</li> <li>• Figure 52: Reworded text about LBAs from ZCAP to ZSZE-1.</li> <li>• Base Spec 5.15.2.12: Changed our modification to align with TP 4056b wording.</li> </ul>
2020-10-27	<p>Changes from 2020-10-27 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• Added “then” for clarity at numerous places.</li> <li>• 4.3.1.4 (Reset Zone): Reorganized wording of Zone Descriptor attribute changes.</li> <li>• Corrected inadvertent renumbering of figures 44 through 52.</li> <li>• Deleted previously-resolved comments.</li> </ul>
2020-11-03	<p>Changes from 2020-11-03 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• Deleted previously-resolved comments.</li> <li>• 4.3.1.4: Added reference to Command Completion section.</li> <li>• 4.3.1.4: Deleted paragraph explaining how to get list of changed zones. This text is already in the Command Completion section.</li> <li>• 4.4: Reworded text in second paragraph.</li> <li>• A.5: Modified text block in Figure 51.</li> <li>• Merged the “TP 4056 CHANGES” section into the “BASE SPEC CHANGES” section.</li> <li>• Base spec: Added “I/O” to all instances of “command sets that specify logical blocks”.</li> <li>• Base spec 5.13: Modified note 1 in Figure 186.</li> <li>• Base spec 5.14: Removed change to this section; it will be added to TP 4056d.</li> <li>• Base spec Figure 348: Removed note 6.</li> </ul>
2020-11-10	<p>Changes from 2020-11-10 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• 2.4: Added clarifying statement that processing of one command may cause two sequential state transitions.</li> <li>• Corrected typo.</li> <li>• Deleted all resolved comments.</li> </ul>
2020-11-17	<p>Changes from 2020-11-12 Technical WG meeting:</p> <ul style="list-style-type: none"> <li>• Removed changes from TP 4068.</li> <li>• Added list of potentially incompatible changes.</li> </ul> <p>Changes from 2020-11-17 meeting of the ZNS Task Group:</p> <ul style="list-style-type: none"> <li>• Reworded the list of Incompatible Changes.</li> </ul>
2020-11-19	<p>Changes from 2020-11-19 Technical WG meeting:</p> <ul style="list-style-type: none"> <li>• Minor rewording of incompatible change list.</li> </ul> <p>This revision is ready for Member Review.</p>
2020-12-21	Added comments received during Member Review
2020-12-22	Comment resolution – first pass
2021-01-05	<p>Changes from 2021-01-05 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• Approved resolutions through section 4.2.7.1.</li> <li>• Accepted changes for the above resolutions.</li> <li>• 4.3: Proposed fix for write protection text.</li> </ul>
2021-01-12	<p>Changes from 2021-01-12 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• 2.5.1: Revised second paragraph and following list. For review in next meeting.</li> <li>• 4.3.1.3: Provided alternative paragraph for the last paragraph in the section.</li> <li>• Figure 39: Clarified the description of the Zone Capacity (ZCAP) field.</li> </ul>
2021-01-19	<p>Changes from 2021-01-19 ZNS TG meeting:</p> <ul style="list-style-type: none"> <li>• 2.5.1: Again revised second paragraph and following list. This resolution is probably final.</li> </ul>

2021-01-26	<p>Added unresolved comments from Jim Hatfield.</p> <p>Changes from 2021-01-26 ZNS TG Meeting:</p> <ul style="list-style-type: none"> <li>• 2.5.1: Final cleanup of handling resource unavailability.</li> <li>• Various sections: Minor grammar changes.</li> <li>• Resolved all open comments (other than notes to editor).</li> <li>• Added "TWG" comments for review in 2021-01-28 Technical WG.</li> </ul>
2021-01-28	<p>Changes after Integration approval by Technical WG on 2021-01-28:</p> <ul style="list-style-type: none"> <li>• Deleted "TWG" comments.</li> </ul>
2021-01-29	<p>First pass integration: Did not modify BASE SPEC CHANGES section.</p> <p>Accepted changes related to figure numbers and section numbers.</p>
2021-02-10	<p>Integrated into the NVMe Base Specification</p>
2021-02-17	<p>Changes from 2021-02-17 Refactoring TG meeting:</p> <ul style="list-style-type: none"> <li>• Align four fields in the SMART / Health Information log page fields and in the Endurance Group Information log page with those in the NVM command set.</li> </ul>
2021-02-18	<p>Changes from 2021-02-18 Technical WG meeting:</p> <ul style="list-style-type: none"> <li>• 3.4: Modified Commands Affected column of Figure 14. Corrected log page name.</li> </ul>
2021-04-06	<p>Integrated into the NVMe Base Specification.</p>

## Description for NVMe Changes Document

This technical proposal defines Zoned Namespaces and the associated Zoned Namespace Command Set. A Zoned Namespace divides the logical address space of a namespace into zones. Each zone is an LBA range that shall be written sequentially, and if rewritten shall be explicitly reset. The interface allows zoned namespaces to expose natural boundaries for internal device structures, and offload management of internal mapping tables to the host.

## Incompatible Changes from TP 4053 to TP 4053a

- The TP did not clearly state that the Identify Namespace data structure (CNS 00h) is supported. The TP now includes it in a list of Identify Namespace data structures (section 3.1).
- The TP did not state which status code is to be returned if the Zone Management Send command Open Zone action is aborted because insufficient resources are available. The TP now states that the status code is Too Many Active Zones or Too Many Open Zones (section 2.5.1).
- The TP did not state whether the Zone Append command supports Directives. The TP now defines that the command contains the DTYPE and DSPEC fields (section 4.5).
- The TP originally specified that the commands affecting the Data Units Written field in the SMART / Health Information log page and affecting the Host Read Commands field in the Endurance Group Information log page are different from the commands that affect those fields in the NVM Command Set. The TP now aligns the behavior with the definitions in the NVM Command Set (section 3.4).

## Description of Specification Changes

Markup Conventions:

Black:	Unchanged (however, hot links are removed)
<del>Red Strikethrough:</del>	Deleted
Blue:	New
Red Highlighted:	TBD values, anchors, and links to be inserted.
<Green Bracketed>:	Notes to editor



**NVM Express™**

**Zoned Namespace**

# **Command Set Specification**

**NVM Express™**

**Revision 1.0**

**Date TBD**

*Please send comments to [info@nvmexpress.org](mailto:info@nvmexpress.org)*

*Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.*

## SPECIFICATION DISCLAIMER

### **LEGAL NOTICE:**

© **Copyright 2021 NVM Express, Inc. ALL RIGHTS RESERVED.**

This NVM Express Zoned Namespace Command Set specification revision 1.0 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 NVM Express Zoned Namespace Command Set specification subject, however, to the Member’s continued compliance with the Company’s Intellectual Property Policy and Bylaws and the Member’s Participation Agreement.

**NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.:** If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: “© 2021 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.

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



## Table of Contents

<b>1</b>	<b>INTRODUCTION .....</b>	<b>11</b>
1.1	Overview.....	11
1.2	Scope .....	11
1.3	Outside of Scope .....	11
1.4	Conventions.....	11
1.5	Definitions .....	11
1.6	References .....	12
1.7	Theory of Operation.....	12
<b>2</b>	<b>ZONED NAMESPACES.....</b>	<b>13</b>
2.1	Introduction .....	13
2.2	Zone Descriptor .....	13
2.3	Zone Types.....	14
2.4	Zone State Machine .....	16
2.5	Zone Resources .....	19
2.6	Logical Block Allocation and Capacity Management .....	20
<b>3</b>	<b>ZONED ADMIN COMMAND SET.....</b>	<b>22</b>
3.1	Identify Namespace Data Structures.....	22
3.2	Identify Controller Data Structures .....	24
3.3	Asynchronous Events.....	25
3.4	Log Pages .....	26
3.5	Admin Commands .....	27
3.6	Controller Architecture.....	28
<b>4</b>	<b>ZONED I/O COMMANDS .....</b>	<b>30</b>
4.1	Command Specific Status Codes.....	30
4.2	NVM Command Set I/O Commands – Zoned Namespace Command Set Specific.....	31
4.3	Zone Management Send command .....	33
4.4	Zone Management Receive command .....	37
4.5	Zone Append command .....	41
4.6	Controller Architecture.....	43
4.7	Reservations.....	44
<b>5</b>	<b>FEATURES.....</b>	<b>45</b>
5.1	Zone Descriptor Extension .....	45
5.2	Reset Zone Recommended .....	45
5.3	Finish Zone Recommended .....	45

5.4	Zone Active Excursions .....	46
-----	------------------------------	----

**ANNEX A. ZONED NAMESPACES HOST CONSIDERATIONS (NORMATIVE) ..... 47**

A.1	Introduction .....	47
A.2	Writing to Zones .....	47
A.3	Open Zone Considerations .....	47
A.4	Partial Failures.....	48
A.5	Capacity and Sizes.....	48

# 1 Introduction

## 1.1 Overview

The NVM Express™ (NVMe™) Zoned Namespace Command Set specification allows host software to communicate with a non-volatile memory subsystem using zones. This specification builds on features defined in the NVM Express Base specification, the PCI Express transport documented in the NVMe™ over PCIe specification, and the Fabric transport documented in the NVMe™ over Fabrics specification.

### 1.1.1 NVMe™ over PCIe™ and NVMe™ over Fabrics

The NVM Express Base specification revision 1.4a and prior revisions define a register level interface for host software to communicate with a non-volatile memory subsystem over PCI Express (NVMe over PCIe specification). The NVMe over Fabrics specification defines a protocol interface and related extensions to the NVMe interface that enable operation over other interconnects (e.g., Ethernet, InfiniBand™, Fibre Channel). The NVMe over Fabrics specification has an NVMe Transport binding for each NVMe Transport (either within that specification or by reference).

In this specification, a requirement/feature may be documented as specific to NVMe over Fabrics implementations or to a particular NVMe Transport binding. In addition, support requirements for features and functionality may differ between NVMe over PCIe and NVMe over Fabrics implementations.

## 1.2 Scope

This specification defines the Zoned Namespace Command Set that enables a host to communicate with a non-volatile memory subsystem using zoned namespaces. This specification supplements the NVMe Base specification.

## 1.3 Outside of Scope

The register interface and command set are specified apart from any usage model for the NVM, but rather only specifies the communication interface to the NVM subsystem. Thus, this specification does not specify whether the non-volatile memory system is used as a solid state drive, a main memory, a cache memory, a backup memory, a redundant memory, etc. Specific usage models are outside the scope, optional, and not licensed.

The implementation or use of other published specifications referred to in this specification, even if required for compliance with the specification, are outside the scope of this specification (e.g., PCI, PCI Express, and PCI-X).

## 1.4 Conventions

This specification uses the conventions defined in the NVMe Base specification.

## 1.5 Definitions

This specification uses the definitions from the NVMe Base specification and the additional definitions in this section.

### 1.5.1 active zone

A zone that is in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state.

### 1.5.2 Address-Specific Write Command

A Write command, Write Zeroes command, Write Uncorrectable command, or Copy command. Address-Specific Write Commands specify a specific logical block range of addresses in command parameters as part of the Submission Queue Entry or in data structures pointed to by the command parameters to be written.

*Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.*

### **1.5.3 open zone**

A zone that is in the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state.

### **1.5.4 read operation**

An operation initiated by a Read, Compare, Verify, or Copy command.

### **1.5.5 write operation**

An operation initiated by a Write, Write Zeroes, Write Uncorrectable, Copy, or Zone Append command.

### **1.5.6 zone**

A contiguous range of logical block addresses that is managed as a single unit.

### **1.5.7 Zone Descriptor**

The data structure that contains information about a zone.

### **1.5.8 Zone Descriptor Extension**

Host defined data that is associated with a zone.

### **1.5.9 zoned namespace**

A namespace that is divided into zones and is associated with the Zoned Namespace Command Set.

## **1.6 References**

The NVM Express Base specification revision 1.4a is referred to as the NVMe Base specification.

## **1.7 Theory of Operation**

The NVM Express host controller interface is designed to address the needs of Enterprise and Client systems that utilize NVM subsystems. The interface utilizes the command submission and completion paths defined in the NVMe Base specification.

This specification along with the NVMe Base specification and transport specific specifications (e.g., NVMe over PCIe, NVMe over Fabrics) defines a streamlined set of registers whose functionality includes:

- a) indication of controller capabilities;
- b) status for controller failures (command status is processed via CQ directly);
- c) Admin Queue configuration (I/O Queue configuration processed via Admin commands); and
- d) doorbell registers for scalable number of Submission and Completion Queues.

This specification defines the Zoned Namespace Command Set.



Attribute	Description																		
Zone State	<div>Each zone has an associated state machine. That state machine has a set of states and each state, together with the zone type, defines the operational characteristics of that zone.</div> <table><tr><th>Zone Type</th><th>Zone State</th><th>Reference Section</th></tr><tr><td rowspan="7">Sequential Write Required</td><td>ZSE:Empty</td><td>2.4.1</td></tr><tr><td>ZSIO:Implicitly Opened</td><td>2.4.2</td></tr><tr><td>ZSEO:Explicitly Opened</td><td>2.4.3</td></tr><tr><td>ZSC:Closed</td><td>2.4.4</td></tr><tr><td>ZSF:Full</td><td>2.4.5</td></tr><tr><td>ZSRO:Read Only</td><td>2.4.6</td></tr><tr><td>ZSO:Offline</td><td>2.4.7</td></tr></table>	Zone Type	Zone State	Reference Section	Sequential Write Required	ZSE:Empty	2.4.1	ZSIO:Implicitly Opened	2.4.2	ZSEO:Explicitly Opened	2.4.3	ZSC:Closed	2.4.4	ZSF:Full	2.4.5	ZSRO:Read Only	2.4.6	ZSO:Offline	2.4.7
Zone Type	Zone State	Reference Section																	
Sequential Write Required	ZSE:Empty	2.4.1																	
	ZSIO:Implicitly Opened	2.4.2																	
	ZSEO:Explicitly Opened	2.4.3																	
	ZSC:Closed	2.4.4																	
	ZSF:Full	2.4.5																	
	ZSRO:Read Only	2.4.6																	
	ZSO:Offline	2.4.7																	
Write Pointer	The Write Pointer attribute defines the next writeable logical block address in that zone. The validity of the write pointer is zone state specific and defined per zone type (refer to section 2.3).																		
Zone Start Logical Block Address	The Zone Start Logical Block Address (ZSLBA) attribute defines the lowest logical block address for that zone.																		
Zone Capacity	The Zone Capacity attribute defines the writable capacity of that zone.																		
Zone Descriptor Extension Valid	The Zone Descriptor Extension Valid attribute defines the validity of the Zone Descriptor Extension data of that zone. Refer to section 5.1.																		
Reset Zone Recommended	The Reset Zone Recommended attribute indicates that the controller recommends that the host resets that zone. Refer to section 5.2.																		
Finish Zone Recommended	The Finish Zone Recommended attribute indicates that the controller recommends that the host finishes that zone. Refer to section 5.3.																		
Zone Finished by Controller	The Zone Finished by Controller attribute indicates that the controller finished that zone due to a Zone Active Excursion. Refer to section 5.4.																		

## 2.3 Zone Types

### 2.3.1 Sequential Write Required Zones

A zone type of Sequential Write Required requires the set of logical block addresses of a zone to be written sequentially.

#### 2.3.1.1 Writing in Sequential Write Required Zones

The following commands may be used to write to logical blocks in a specific zone of zone type Sequential Write Required: Write, Write Zeroes, Write Uncorrectable, Copy, and Zone Append.

A write pointer is maintained for each zone in the zoned namespace that indicates the next writeable logical block address in that zone. The write pointer is valid for a subset of the zone states as defined in Figure 3.

**Figure 3: Zone Characteristics**

State	Zone Characteristics		
	Valid Write Pointer <sup>1</sup>	Active Resources <sup>2</sup>	Open Resources <sup>2</sup>
ZSE:Empty	Yes	No	No
ZSIO:Implicitly Opened	Yes	Yes	Yes
ZSEO:Explicitly Opened	Yes	Yes	Yes
ZSC:Closed	Yes	Yes	No
ZSF:Full	No	No	No
ZSRO:Read Only	No	No	No
ZSO:Offline	No	No	No

State	Zone Characteristics		
	Valid Write Pointer <sup>1</sup>	Active Resources <sup>2</sup>	Open Resources <sup>2</sup>

NOTES:

1. A valid write pointer (i.e., Yes) indicates that the write pointer zone attribute within that zone contains a valid logical block address. An invalid write pointer (i.e., No) provides no information.
2. Resources associated with a zone are defined in section 2.5.

The host may use the Zone Management Receive command to determine the current write pointer for a zone.

The write pointer for a zone in the ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state shall be increased by the number of logical blocks written on successful completion of a write operation.

If the controller is not able to successfully write to all logical blocks specified by a write operation, then the write pointer shall:

- a) be set to a value within the range of LBAs specified in that write operation;
- b) be set to one greater than the last LBA in the range of LBAs specified in that write operation; or
- c) become invalid (i.e., due to transitioning to the ZSRO:Read Only state or the ZSO:Offline state, or due to a Zone Active Excursion (refer to section 5.4)).

The Zone Management Send command with Zone Send Action of Reset Zone sets the write pointer to the ZSLBA for that zone.

Figure 4 shows an example of a zone in the ZSE:Empty state LBA  $m$  is the ZSLBA attribute, the write pointer indicates ZSLBA, and  $n-1$  is the highest LBA of the zone.

**Figure 4: Write Pointer in an Empty Zone**

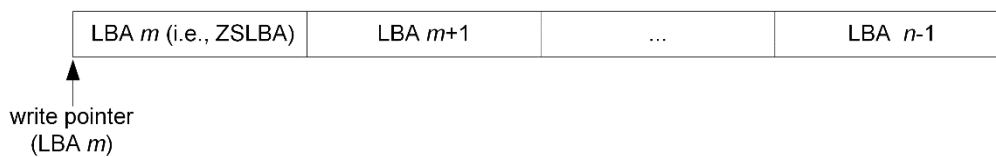
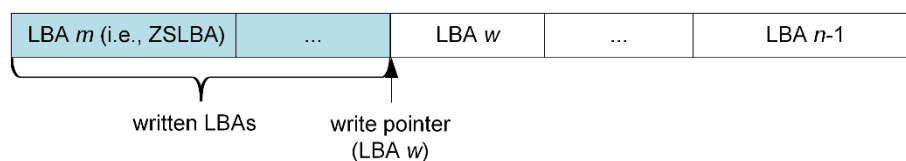


Figure 5 shows an example of a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state, that has had some logical blocks written (refer to Figure 5). The write pointer, indicated by LBA  $w$ , is the lowest-numbered unwritten LBA (i.e., the next LBA to be written) and  $n-1$  is the highest LBA of the zone.

**Figure 5: Write Pointer in a Partially Written Zone**



The controller shall abort a command that writes to a zone that is in the ZSF:Full state with a status code of Zone Is Full.

The controller shall abort a command that writes to a zone that is in the ZSRO:Read Only state with a status code of Zone Is Read Only.

The controller shall abort a command that writes to a zone that is in the ZSO:Offline state with a status code of Zone Is Offline.

For a zone in the ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state:

- a) if an Address-Specific Write Command specifies a Starting LBA field that is not equal to the write pointer for that zone, then the controller shall abort that command with a status code of Zone Invalid Write.
- b) if a Zone Append command specifies a ZSLBA that is not the lowest logical block address in that zone, then the controller shall abort that command with a status code of Invalid Field in Command.

The controller shall abort a command that initiates a write operation that has a starting LBA in one zone for which the Number of Logical blocks exceeds the remaining number of logical blocks in that zone, with a status code of Zone Boundary Error.

The Command Ordering Requirements section in the NVMe Base specification does not guarantee ordering of commands being processed by the controller. If an Address-Specific Write Command is submitted for a zone while an Address-Specific Write Command or a Zone Append command is outstanding for that zone, then the controller may process an Address-Specific Write Command that specifies a Starting LBA field that does not equal the write pointer for that zone, due to the order of processing. If the controller processes an Address-Specific Write Command that specifies a Starting LBA field that is not equal to the write pointer for the associated zone, then the controller aborts that command as described in this section. Multiple outstanding Zone Append commands do not cause this error.

### **2.3.1.2 Reading in Sequential Write Required Zones**

If a command specifies a read operation on logical blocks within a zone, and the zone is:

- a) in the ZSO:Offline state, then that command shall be aborted with a status code of Zone Is Offline; or
- b) in any other state, then that read operation is performed as described in this section, in section 2.6, and in the Deallocate section in the NVMe Base specification.

If the Read Across Zone Boundaries bit in the Zoned Namespace Command Set specific Identify Namespace data structure (refer to Figure 8) is set to '1', then commands are allowed to perform read operations that specify an LBA range containing logical blocks in more than one zone.

If the Read Across Zone Boundaries bit is cleared to '0', then commands that perform read operations that specify an LBA range containing logical blocks in more than one zone shall be aborted with a status code of Zone Boundary Error.

## **2.4 Zone State Machine**

There is a state machine associated with each zone. The state machine controls the operational characteristics of each zone. The state machine consists of the following states: ZSE:Empty, ZSIO:Implicitly Opened, ZSEO:Explicitly Opened, ZSC:Closed, ZSF:Full, ZSRO:Read Only, and ZSO:Offline.

The initial state of a zone state machine is set as a result of:

- a) an NVM Subsystem Reset; or
- b) all controllers in the NVM subsystem reporting Shutdown processing complete (i.e., 10b in the Shutdown Status (SHST) register, refer to the NVMe Base specification).

The initial state for each zone is the:

- a) ZSE:Empty state:
  - a) if the write pointer is valid, the write pointer points to the lowest LBA in the zone, and the Zone Descriptor Extension Valid bit is cleared to '0';
- b) ZSC:Closed state:
  - a) if the write pointer is valid and does not point to the lowest LBA in the zone; or
  - b) If the write pointer is valid and the Zone Descriptor Extension Valid bit is set to '1';
- c) ZSF:Full state:
  - a) if the most recent state was the ZSF:Full; or



- b) if the zone state was transitioned to the ZSF:Full state as a result of the NVM Subsystem Reset;
- d) ZSRO:Read Only state, if the most recent zone state was the ZSRO:Read Only state; and
- e) ZSO:Offline state, if the most recent zone state was the ZSO:Offline state.

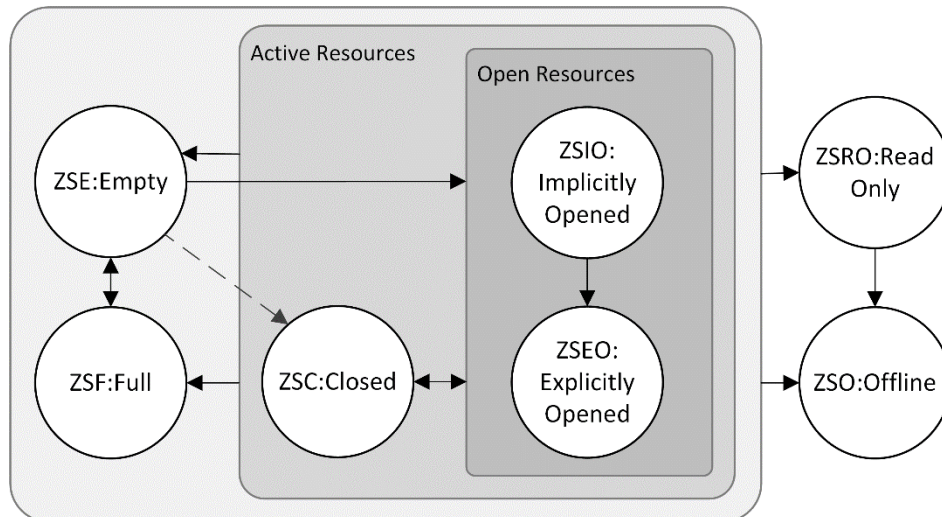
Transitions between zone states cause reporting of a Zone Descriptor Changed asynchronous event as described in section 3.3.1.

If the zoned namespace is formatted with a Format NVM command or created with a Namespace Management command, the zones in the zoned namespace are initialized to either the ZSE:Empty state or the ZSO:Offline state.

Zones that are in the ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state may transition to the ZSRO:Read Only state or ZSO:Offline state by mechanisms outside the scope of this specification.

Figure 6 shows the valid transitions between each zone state from the time a zoned namespace is formatted or created. The transition (dotted line) from the ZSE:Empty state to the ZSC:Closed state is only valid when the zoned namespace is formatted with Zone Descriptor Extension support (refer to section 5.1). An arrow to or from a shaded area indicates transitions to or from all states in that area.

**Figure 6: Zone State Machine**



The processing of a command may cause multiple state machine transitions (e.g., the processing of a Write command may cause a transition from the ZSC:Closed state to the ZSIO:Implicitly Opened state and then cause a transition from the ZSIO:Implicitly Opened state to the ZSF:Full state).

#### 2.4.1 ZSE:Empty state

**Transition ZSE:ZSIO:** The zone shall transition from the ZSE:Empty state to the ZSIO:Implicitly Opened state if there are available resources as defined in section 2.5 and a write operation writes one or more logical blocks of that zone.

**Transition ZSE:ZSEO:** The zone shall transition from the ZSE:Empty state to the ZSEO:Explicitly Opened state if there are available resources as defined in section 2.5 and a Zone Management Send command with a Zone Send Action of Open Zone completes successfully.

**Transition ZSE:ZSC:** The zone shall transition from the ZSE:Empty state to the ZSC:Closed state if there are available active resources as defined in section 2.5 and a Zone Management Send command with a Zone Send Action of Set Zone Descriptor Extension completes successfully.

**Transition ZSE:ZSF:** The zone shall transition from the ZSE:Empty state to the ZSF:Full state as a result of a Zone Management Send command with a Zone Send Action of Finish Zone completes successfully.

## 2.4.2 ZSIO:Implicitly Opened state

**Transition ZSIO:ZSE:** The zone shall transition from the ZSIO:Implicitly Opened state to the ZSE:Empty state as a result of successful completion of a Zone Management Send command with a Zone Send Action of Reset Zone.

**Transition ZSIO:ZSEO:** The zone shall transition from the ZSIO:Implicitly Opened state to the ZSEO:Explicitly Opened state as a result of successful completion of a Zone Management Send command with a Zone Send Action of Open Zone.

**Transition ZSIO:ZSC:** The zone shall transition from the ZSIO:Implicitly Opened state to the ZSC:Closed state as a result of:

- a) successful completion of a Zone Management Send command with a Zone Send Action of Close Zone; and
- b) the controller initiating the transition as described in section 2.5.1.

**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 5.4); and
- d) as a result of the zoned namespace becomes write protected (refer to the Namespace Write Protection section in the NVMe Base specification).

## 2.4.3 ZSEO:Explicitly Opened state

**Transition ZSEO:ZSE:** The zone shall transition from the ZSEO:Explicitly Opened state to the ZSE:Empty state as a result of successful completion of a Zone Management Send command with a Zone Send Action of Reset Zone.

**Transition ZSEO:ZSC:** A zone shall transition from the ZSEO:Explicitly Opened state to the ZSC:Closed state upon successful completion of a Zone Management Send command with a Zone Send Action of Close Zone.

**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 5.4); or
- d) as a result of the zoned namespace becomes write protected (refer to the Namespace Write Protection section in the NVMe Base specification).

## 2.4.4 ZSC:Closed state

**Transition ZSC:ZSE:** The zone shall transition from the ZSC:Closed state to the ZSE:Empty state as a result of successful completion of a Zone Management Send command with a Zone Send Action of Reset Zone.

**Transition ZSC:ZSIO:** The zone shall transition from the ZSC:Closed state to the ZSIO:Implicitly Opened state, if there are available resources as defined in section 2.5 and a write operation that writes one or more logical blocks to the zone completes successfully.

**Transition ZSC:ZSEO:** The zone shall transition from the ZSC:Closed state to the ZSEO:Explicitly Opened state if there are available resources as defined in section 2.5 and a Zone Management Send command with a Zone Send Action of Open Zone completes successfully.

**Transition ZSC:ZSF:** The zone shall transition from the ZSC:Closed 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) due to a Zone Active Excursion (refer to section 5.4); and
- c) as a result of the zoned namespace becoming write protected (refer to the Namespace Write Protection section in the NVMe Base specification).

#### 2.4.5 ZSF:Full state

**Transition ZSF:ZSE:** The zone shall transition from the ZSF:Full state to the ZSE:Empty state as a result of successful completion of a Zone Management Send command with a Zone Send Action of Reset Zone.

#### 2.4.6 ZSRO:Read Only

**Transition ZSRO:ZSO:** The zone shall transition from the ZSRO:Read Only state to the ZSO:Offline state as a result of successful completion of a Zone Management Send command with a Zone Send Action of Offline Zone.

#### 2.4.7 ZSO:Offline state

There are no transitions from the ZSO:Offline state to any other zone state.

### 2.5 Zone Resources

Zones may have associated Active Resources and Open Resources attached. The resources limit the number of zones that are allowed to be in each zone state. The resource relationship is defined in Figure 7.

**Figure 7: Zone Resources**

Resource	States	Comment
Active	ZSIO:Implicitly Opened, ZSEO:Explicitly Opened, ZSC:Closed	Zones in zone states associated with this resource are limited by the Maximum Active Resources field.
Open	ZSIO:Implicitly Opened, ZSEO:Explicitly Opened	Zones in zone states associated with this resource are limited by the Maximum Open Resources field.

Zones that have associated Open Resources are a subset of the zones that are associated with Active Resources. The Maximum Open Resources field shall be less than or equal to the Maximum Active Resources field.

#### 2.5.1 Managing resources

The controller associates Active Resources with zones in the ZSIO:Implicitly Opened, the ZSEO:Explicitly Opened, and the ZSC:Closed states. The controller associates Open Resources with zones in the ZSIO:Implicitly Opened and the ZSEO:Explicitly Opened states.

The resource management is as follows:

- a) A transition from the ZSE:Empty state to the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state increases the resource usage of Active Resources and Open Resources by 1.
- b) A transition from the ZSE:Empty state to the ZSC:Closed state increases the resource usage of Active Resources by 1.
- c) A transition from the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state to the ZSC:Closed state decreases the resource usage of Open Resources by 1.
- d) A transition from the ZSC:Closed state to the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state increases the resource usage of Open Resources by 1.

- e) A transition from the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state to the ZSE:Empty state, the ZSF:Full state, the ZSRO:Read Only state, or the ZSO:Offline state decreases the resource usage of Active Resources and Open Resources by 1.
- f) A transition from the ZSC:Closed state to the ZSE:Empty state, the ZSF:Full state, the ZSRO:Read Only state, or the ZSO:Offline state decreases the resource usage of Active Resources by 1.
- g) A transition from the ZSE:Empty state to the ZSF:Full state shall not impact resource usage of Active Resources and Open Resources.

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

- a) the Maximum Active Resources field is greater than the Maximum Open Resources field and:
  - i. as a result of the requested transition the resource that is not available is Active Resources, then abort the command with a status code of Too Many Active Zones; or
  - ii. as a result of the requested transition the resource that is not available is Open Resources, then abort the command with a status code of Too Many Open Zones;

or

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

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

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

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

## 2.6 Logical Block Allocation and Capacity Management

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

- a) a Write command;
- b) a Write Uncorrectable command;
- c) a Write Zeroes command that does not deallocate the logical block (refer to the Deallocate section in the NVMe Base specification);
- d) a Copy command; and
- e) a Zone Append command.

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

- a) a Write command not addressing that logical block (e.g., a write of LBA n causes allocation of LBAs n and n+1);
- b) a Write Uncorrectable command not addressing that logical block;
- c) a Write Zeroes command not addressing that logical block (refer to the Write Zeroes command in the NVMe Base specification);
- d) a Copy command not addressing that logical block;
- e) a Zone Management Send command that transitions a zone containing that logical block to the ZSEO:Explicitly Opened state or the ZSC:Closed state; and
- f) a Zone Append command.

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

- a) a Dataset Management command;
- b) a Write Zeroes command addressing that deallocates logical blocks (refer to the Deallocate section in the NVMe Base specification); and
- c) a sanitize operation.

All logical blocks in a zone shall be marked as deallocated when:

- a) the zone is in the ZSE:Empty state or the ZSO:Offline state.

### 3 Zoned Admin Command Set

This specification supports commands in the Admin Command Set and the NVM Command Set Specific Admin Commands as described in the NVMe Base specification. Admin Commands that are specific to the Zoned Namespace Command Set specification are defined in this section.

#### 3.1 Identify Namespace Data Structures

The Zoned Namespace Command Set supports the:

- Identify Namespace data structure (CNS 00h) (refer to the NVMe Base specification);
- Identify I/O Command Set specific Namespace data structure for the NVM Command Set (CNS 05h, CSI 00h) (refer to the NVM Command Set specification);
- Identify I/O Command Set specific Identify Namespace data structure for the Zoned Namespace Command Set (CNS 05h, CSI 02h) (refer to Figure 8); and
- I/O Command Set Independent Identify Namespace data structure (CNS 08h) (refer to NVMe Base specification).

**Figure 8: Identify I/O Command Set Specific Namespace Data Structure, Zoned Namespace Command Set**

Bytes	O/M <sup>1</sup>	Description								
01:00	O	<b>Zone Operation Characteristics (ZOC):</b> This field indicates the zone operation characteristics of the zoned namespace.								
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>15:2</td><td>Reserved</td></tr><tr><td>1</td><td><b>Zone Active Excursions:</b> If set to '1', then a controller may transition a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state to the ZSF:Full state due to a vendor specific excursion event. If cleared to '0', then a controller shall not transition a zone due to a vendor specific excursion event. Refer to section 5.4.</td></tr><tr><td>0</td><td><b>Variable Zone Capacity:</b> if set to '1', then the capacity for a zone may change without a change to the format of the zoned namespace. If cleared to '0', then the capacity for a zone does not change without a change to the format of the zoned namespace. Refer to Figure 39.</td></tr></table>	Bits	Description	15:2	Reserved	1	<b>Zone Active Excursions:</b> If set to '1', then a controller may transition a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state to the ZSF:Full state due to a vendor specific excursion event. If cleared to '0', then a controller shall not transition a zone due to a vendor specific excursion event. Refer to section 5.4.	0	<b>Variable Zone Capacity:</b> if set to '1', then the capacity for a zone may change without a change to the format of the zoned namespace. If cleared to '0', then the capacity for a zone does not change without a change to the format of the zoned namespace. Refer to Figure 39.
		Bits	Description							
		15:2	Reserved							
1	<b>Zone Active Excursions:</b> If set to '1', then a controller may transition a zone in the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state to the ZSF:Full state due to a vendor specific excursion event. If cleared to '0', then a controller shall not transition a zone due to a vendor specific excursion event. Refer to section 5.4.									
0	<b>Variable Zone Capacity:</b> if set to '1', then the capacity for a zone may change without a change to the format of the zoned namespace. If cleared to '0', then the capacity for a zone does not change without a change to the format of the zoned namespace. Refer to Figure 39.									
03:020	O	<b>Optional Zoned Command Support (OZCS):</b> This field defines optional features of the zoned namespace.								
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>15:1</td><td>Reserved</td></tr><tr><td>0</td><td><b>Read Across Zone Boundaries:</b> If set to '1', then any command is allowed to perform read operations that specify an LBA range containing logical blocks in more than one zone.  If cleared to '0', then any command that performs a read operation that specifies an LBA range containing logical blocks in more than one zone is aborted as described in section 2.3.1.2.</td></tr></table>	Bits	Description	15:1	Reserved	0	<b>Read Across Zone Boundaries:</b> If set to '1', then any command is allowed to perform read operations that specify an LBA range containing logical blocks in more than one zone.  If cleared to '0', then any command that performs a read operation that specifies an LBA range containing logical blocks in more than one zone is aborted as described in section 2.3.1.2.		
		Bits	Description							
15:1	Reserved									
0	<b>Read Across Zone Boundaries:</b> If set to '1', then any command is allowed to perform read operations that specify an LBA range containing logical blocks in more than one zone.  If cleared to '0', then any command that performs a read operation that specifies an LBA range containing logical blocks in more than one zone is aborted as described in section 2.3.1.2.									
07:04	M	<b>Maximum Active Resources (MAR):</b> This field defines the maximum number of concurrently active zones in the zoned namespace. A value of FFFFFFFFh indicates that there is no limit. This is a 0's based value.								
11:08	M	<b>Maximum Open Resources (MOR):</b> This field defines the maximum number of concurrently open zones in the zoned namespace. This field shall be less than or equal to the Maximum Active Resources field. A value of FFFFFFFFh indicates that there is no limit. This is a 0's based value.								
15:12	O	<b>Reset Recommended Limit (RRL):</b> Number of seconds before the NVM subsystem may perform a vendor specific action on a zone after the Reset Zone Recommended zone attribute is set to '1' for that zone. If this field is cleared to 0h, then no Reset Recommended Limit is reported. Refer to section 5.2.								
19:16	O	<b>Finish Recommended Limit (FRL):</b> Number of seconds before the NVM subsystem may perform the vendor specific action on a zone after the Finish Zone Recommended zone attribute is set to '1' for that zone. If this field is cleared to 0h, then no Finish Recommended Limit is reported. Refer to section 5.3.								
2815:20		Reserved								

**Figure 8: Identify I/O Command Set Specific Namespace Data Structure, Zoned Namespace Command Set**

Bytes	O/M <sup>1</sup>	Description
2831:2816	M	<b>LBA Format 0 Extension (LBAFE0):</b> This field indicates the LBA format Extension 0 that is supported by the controller. The Zone format field is defined in Figure 9.
2847:2832	O	<b>LBA Format 1 Extension (LBAFE1):</b> This field indicates the LBA format 1 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
...		
2863:2848	O	<b>LBA Format 2 Extension (LBAFE2):</b> This field indicates the LBA format 2 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2879:2864	O	<b>LBA Format 3 Extension (LBAFE3):</b> This field indicates the LBA format 3 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2895:2880	O	<b>LBA Format 4 Extension (LBAFE4):</b> This field indicates the LBA format 4 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2911:2896	O	<b>LBA Format 5 Extension (LBAFE5):</b> This field indicates the LBA format 5 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2927:2912	O	<b>LBA Format 6 Extension (LBAFE6):</b> This field indicates the LBA format 6 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2943:2928	O	<b>LBA Format 7 Extension (LBAFE7):</b> This field indicates the LBA format 7 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2959:2944	O	<b>LBA Format 8 Extension (LBAFE8):</b> This field indicates the LBA format 8 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2975:2960	O	<b>LBA Format 9 Extension (LBAFE9):</b> This field indicates the LBA format 9 Extension that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
2991:2976	O	<b>LBA Format 10 Extension (LBAFE10):</b> This field indicates the LBA format Extension 10 that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
3007:2992	O	<b>LBA Format 11 Extension (LBAFE11):</b> This field indicates the LBA format Extension 11 that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
3023:3008	O	<b>LBA Format 12 Extension (LBAFE12):</b> This field indicates the LBA format Extension 12 that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
3039:3024	O	<b>LBA Format 13 Extension (LBAFE13):</b> This field indicates the LBA format Extension 13 that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
3055:3040	O	<b>LBA Format 14 Extension (LBAFE14):</b> This field indicates the LBA format Extension 14 that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
3071:3056	O	<b>LBA Format 15 Extension (LBAFE15):</b> This field indicates the LBA format Extension 15 that is supported by the controller. The LBA Format Extension field is defined in Figure 9.
3839:3072		Reserved
4095:3840	O	Vendor Specific
NOTES:		
1. O/M definition: O = Optional, M = Mandatory.		



Figure 9 defines the Identify LBA Format Extension data structure.

**Figure 9: Identify – LBA Format Extension Data Structure**

Bits	Description
127:72	Reserved
71:64	<b>Zone Descriptor Extension Size (ZDES):</b> This field indicates the Zone Descriptor Extension size for each zone. The value is reported in 64B units (e.g., 1h corresponds to 64B, 2h corresponds to 128B). A value of 0h indicates that Zone Descriptor Extensions are not supported. Refer to section 5.1.
63:00	<b>Zone Size (ZSZE):</b> This field contains the size of each zone in the zoned namespace. The value is reported as a number of logical blocks. The value of the field shall not be cleared to 0h.

The LBA Format data structure, NVM Command Set Specific data structure is as defined in the NVMe Base specification. The data structure is extended through the LBA Format Extension data structure.

Commands (e.g., Format NVM and Namespace Management) that use an index to refer to the LBA Format data structure in the NVM Command Set specific Identify Namespace data structure, shall use that same index to refer to the LBA Format Extension data structure and shall process both the LBA Format data structure and LBA Format Extension data structure when an LBA Format is specified.

The host specified namespace management fields are as defined for the NVM Command Set in the NVMe Base specification.

### 3.2 Identify Controller Data Structures

The Zoned Namespace Command Set supports the:

- Identify Controller data structure (CNS 01h) (refer to section 3.2.1);
- Identify Controller data structure for the NVM Command Set (CNS 06h, CSI 00h) (refer to the NVM Command Set specification); and
- Identify Controller data structure for the Zoned Namespace Command Set (CNS 06h, CSI 02h) (refer to section 3.2.2).

#### 3.2.1 Identify Controller Data Structure (CNS 01h)

Figure 10 defines the Zoned Namespace Command Set specific Optional Asynchronous Events Supported field within the Identify Controller data structure (refer to the NVMe Base specification).

**Figure 10: Optional Asynchronous Events Supported (OAES) field, Zoned Namespace Command Set**

Field	Description
Optional Asynchronous Events Supported (OAES)	Bit 27 is set to '1' if the controller supports the Zone Descriptor Changed Notices event and the associated Changed Zone List log page. If cleared to '0', then the controller does not support the Zone Descriptor Changed Notices event nor the associated Changed Zone List log page.

#### 3.2.2 Identify Controller Data Structure, Zoned Namespace Command Set Specific (CNS 06h, CSI 02h)

Figure 11 defines the Zoned Namespace Command Set specific Identify Controller data structure.

**Figure 11: Identify I/O Command Set Specific Controller Data Structure, Zoned Namespace Command Set**

Bytes	O/M <sup>1</sup>	Description
00	0	<b>Zone Append Size Limit (ZASL):</b> If the Zone Append command is supported then: <ol style="list-style-type: none"> <li>a non-zero value in this field indicates the maximum data transfer size for the Zone Append command (refer to section 4.5); and</li> </ol>



Bytes	O/M <sup>1</sup>	Description
		b) a value of 0h in this field indicates that the maximum data transfer size for the Zone Append command is indicated by the Maximum Data Transfer Size (MDTS) field (refer to NVMe Base specification). The value is in units of the minimum memory page size (CAP.MPSMIN) and is reported as a power of two (2^n). This field includes the length of metadata if metadata is interleaved with the stored logical block data. The value of this field shall be less than or equal to the Maximum Data Transfer Size (MDTS).
4095:01		Reserved
NOTES: 1. O/M definition: O = Optional, M = Mandatory.		

### 3.3 Asynchronous Events

This specification supports asynchronous events as described in the NVMe Base specification. Asynchronous events that are specific to the Zoned Namespace Command Set specification are defined in this section.

#### 3.3.1 Zone Descriptor Changed

Figure 12 defines the Zoned Namespace Command Set specific Asynchronous Event Information – Notice data structure.

**Figure 12: Asynchronous Event Information – Notice, Zoned Namespace Command Set**

Value	Description
EFh	<p><b>Zone Descriptor Changed:</b> The Zone Descriptor data structure for a zone changed in a specific zoned namespace. The Zone Descriptor of the zone is indicated in the Changed Zone List log page. To clear this event, host software reads the Zone Changed List log 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 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"> <li>a) a Zone Management Send command that specified that zone;</li> <li>b) a Zone Management Send command that specified all zones;</li> <li>c) a write operation that transitioned that zone:               <ul style="list-style-type: none"> <li>a) from the ZSE:Empty state to the ZSIO:Implicitly Opened state;</li> <li>b) from the ZSIO:Implicitly Opened state to the ZSF:Full state;</li> <li>c) from the ZSEO:Explicitly Opened state to the ZSF:Full state; and</li> <li>d) from the ZSC:Closed state to the ZSIO:Implicitly Opened state;</li> </ul> </li> <li>d) the controller transitioning that zone to the ZSF:Full state due to an NVM Subsystem Reset; and</li> <li>e) the controller transitioning that zone to the ZSC:Closed state.</li> </ul>

##### 3.3.1.1 Command Completion

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

**Figure 13: Asynchronous Event Request – Completion Queue Entry Dword 1**

Bytes	Description
3:0	<b>Namespace Identifier (NSID):</b> This field indicates the namespace identifier that the asynchronous event occurred on.

### 3.4 Log Pages

This specification supports log pages as described in the NVMe Base specification. Log pages that are specific to the Zoned Namespace Command Set specification are defined in this section.

The Commands Supported and Effects log page is mandatory for the Zoned Namespace Command Set.

The log pages defined in the NVMe Base specification contains a set of log page I/O Command Set specific fields. The fields are defined in Figure 14.

**Figure 14: Zoned Namespace Command Set Specific Base Log Pages Fields**

Log Page Name	Field Name	Commands Affected
SMART / Health Information Log	Data Units Read	Read, Compare, Verify
	Data Units Written	Write, <del>Write-Zeroes</del> , Zone Append
	Host Read Commands	Read, Compare, Copy
	Host Write Commands	Write, Zone Append, Copy
Endurance Group Information Log	Host Read Commands	Read, Compare, <del>Verify</del> , Copy
	Host Write Commands	Write, Zone Append, Copy

#### 3.4.1 Log Specific Information

The Zoned Namespace Command Set specific log pages, and log pages defined in other NVM Express specifications and modified by the Zoned Namespace Command Set are shown in Figure 15.

**Figure 15: Get Log Page – Log Page Identifiers, Zoned Namespace Command Set**

Log Identifier	Scope	Log Page Name	Reference Section
BFh	Namespace <sup>1</sup>	Changed Zone List	3.4.1.1
NOTES:			
1. For namespace identifiers other than 0h or FFFFFFFFh.			

##### 3.4.1.1 Changed Zone List (Log Identifier BFh)

This log page indicates if a Zone Descriptor Changed event has occurred for one or more zones. If there is an enabled Zone Descriptor Changed Event pending for a specific zone, then the Changed Zone List includes an entry for that zone.

The Changed Zone List log page is 4,096 bytes in size. It contains a Zone Identifier List with up to 511 Zone Identifier entries. The Zone Identifier List is defined in Figure 16.

A Zone Identifier entry contains the ZSLBA of the zone associated with the changed Zone Descriptor. Each ZSLBA shall appear not more than once in the Zone Identifier List. Unused entries shall be zero-filled.

**Figure 16: Zone Identifier List Data Structure**

Bytes	Description
01:00	<b>Number of Zone Identifiers:</b> This field indicates the number of Zone Identifiers entries in the list.
07:02	Reserved
15:08	<b>Zone Identifier 0:</b> This field contains the ZSLBA of the first zone in the list, if present.
23:16	<b>Zone Identifier 1:</b> This field contains the ZSLBA of the second zone in the list, if present.
...	...
(N*8+15):(N*8+8)	<b>Zone Identifier N:</b> This field contains the ZSLBA of the N+1 zone in the list, if present.

The controller shall set the Number of Zone Identifiers field in the log page to the number of valid entries that follow.

To determine changes, the host reads the Changed Zone List log page, and for each ZSLBA specified in a Zone Identifier entry, issues a Zone Management Receive command with the Zone Receive Action set to Report Zones or Extended Report Zones, and specifying the ZSLBA from that Zone Identifier entry.

The host should read the entire page with the RAE bit cleared to '0'. This page is dynamic and when reading with offset without the header then the data may be changed from a previous read of the page.

For a given zoned namespace, if the controller is unable to populate the Changed Zone List log page with any valid entries and there is at least one Zone Descriptor Changed event since the last time the log page was read, then the controller shall set the Number of Zone Identifiers in that log page to FFFFh, and the remainder of the list shall be zero filled. The host may read the entire Report Zones data structure or Extended Report Zones data structure to discover which Zone Descriptors have changed since the last time the information was read.

The log page content should not be retained after an NVM Subsystem Reset.

### 3.5 Admin Commands

#### 3.5.1 Set Features command

Figure 17 defines the Zoned Namespace Command Set specific Set Features Command Asynchronous Event Configuration – Command Dword 11 data structure (refer to the NVMe Base specification).

**Figure 17: Asynchronous Event Configuration – Command Dword 11**

Bit	Description
27	<b>Zone Descriptor Changed Notices:</b> This bit specifies whether an asynchronous event notification is sent to the host for a Zone Descriptor Changed event. If this bit is set to '1', then the Zone Descriptor Changed event is sent as described in section 3.3.1 when this condition occurs. If this bit is cleared to '0', then the controller shall not send the Zone Descriptor Changed event to the host.

#### 3.5.2 Sanitize command

The Sanitize command is defined in the NVMe Base specification, with additional requirements for controllers implementing the Zoned Namespace Command Set described in this section. The Sanitize Status (SSTAT) (refer to Sanitize Status (Log Identifier 81h) section in the NVMe Base specification) and resultant zone state as a result of a successful sanitize operation of a zoned namespace that supports the Zoned Namespace Command Set are described in Figure 18.

All fields of the Sanitize command are as specified in the NVMe Base specification.

The resulting zone state and event to report as a result of a successful sanitize operation is dependent on the setting of the:

- No Deallocate After Sanitize bit in the Sanitize command that requested the sanitize operation;
- No-Deallocate Modifies Media After Sanitize field;
- No-Deallocate Inhibited bit;
- No-Deallocate Response Mode bit; and
- Sanitize Action field.

**Figure 18: Sanitize Behavior for the Zoned Namespace Command Set**

No Deallocate After Sanitize	No-Deallocate Modifies Media After Sanitize (NODMMAS)	No-Deallocate Inhibited (NDI)	No-Deallocate Response Mode (NODRM)	Results of a successful sanitize operation		
				Zone State <sup>1</sup>	Logical Block Content <sup>2</sup>	Sanitize Status <sup>4</sup>
0b	xxb <sup>3</sup>	xb <sup>3</sup>	N/A <sup>3</sup>	ZSE:Empty	Refer to section 2.3.1.2	001b
1b	00b	0b	N/A <sup>3</sup>	ZSE:Empty	Refer to section 2.3.1.2	001b
1b	00b	1b	0b			N/A <sup>5</sup>
1b	00b	1b	1b	ZSE:Empty	Refer to section 2.3.1.2	001b

No Deallocate After Sanitize	No-Deallocate Modifies Media After Sanitize (NODMMAS)	No-Deallocate Inhibited (NDI)	No-Deallocate Response Mode (NODRM)	Results of a successful sanitize operation		
				Zone State <sup>1</sup>	Logical Block Content <sup>2</sup>	Sanitize Status <sup>4</sup>
1b	01b (does not modify)	0b (not inhibited)	N/A <sup>3</sup>	ZSF:Full	Block Erase: not defined in this specification  Crypto Erase: not defined in this specification  Overwrite: Shall be the overwrite pattern specified in the Sanitize command that requested the sanitize operation	001b
1b	01b	1b (inhibited)	1b	ZSE:Empty	Refer to section 2.3.1.2	100b
1b	01b	1b	0b			N/A <sup>5</sup>
1b	10b (does modify)	0b	N/A <sup>3</sup>	ZSF:Full	Block Erase: not defined in this specification  Crypto Erase: not defined in this specification  Overwrite: Overwrite: Shall be the overwrite pattern specified in the Sanitize command that requested the sanitize operation	001b
1b	10b	1b	1b	ZSE:Empty	Refer to section 2.3.1.2	100b
1b	10b	1b	0b			N/A <sup>5</sup>
1b	11b	N/A <sup>3</sup>	N/A <sup>3</sup>			N/A <sup>5</sup>

**NOTES:**

1. ZSO:Offline state is a valid zone state as a result of a successful sanitize operation.
2. This field describes the read value from a deallocated logical block. Refer to the Deallocate section in the NVMe Base specification.
3. N/A means that value is not relevant in the setup conditions described in that row.
4. Value reported in bits 2:0 of the Sanitize Status (SSTAT) field in the Sanitize Status (Log Identifier 81h) (refer to the NVMe Base specification).
5. Sanitize command is aborted with a status code of Invalid Field in Command.

### 3.6 Controller Architecture

This specification implements the administrative controller architecture as defined in the NVMe Base specification, with the following additions:

### 3.6.1 Administrative Controller

This specification implements the Administrative – Controller Log Page Support (refer to the NVMe Base specification). Figure 19 defines the Zoned Namespace Command Set specific log page support requirements.

**Figure 19: Administrative – Controller Log Page Support, Zoned Namespace Command Set**

Log Page Name	Command Support Requirements <sup>1</sup>
Changed Zone List	P
NOTES:	
1. O = Optional, M = Mandatory, P = Prohibited	

## 4 Zoned I/O Commands

This specification supports the NVM Command Set commands as described in the NVMe Base specification. Commands that are specific to the Zoned Namespace Command Set specification are defined in this section.

Figure 20 defines the Zoned Namespace Command Set specific commands.

**Figure 20: Opcodes for Zoned Namespace Command Set I/O Commands**

Opcode by Field			Combined Opcode <sup>1</sup>	Command <sup>2</sup>	Reference
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer <sup>3</sup>			
0b	000 00b	00b	00h	Flush <sup>4</sup>	NVMe Base specification
0b	000 00b	01b	01h	Write	4.2.1 and NVMe Base specification
0b	000 00b	10b	02h	Read	4.2.2 and NVMe Base specification
0b	000 01b	00b	04h	Write Uncorrectable	4.2.3 and NVMe Base specification
0b	000 01b	01b	05h	Compare	4.2.4 and NVMe Base specification
0b	000 10b	00b	08h	Write Zeroes	4.2.5 and NVMe Base specification
0b	000 10b	01b	09h	Dataset Management	NVMe Base specification
0b	000 11b	00b	0Ch	Verify	4.2.6 and NVMe Base specification
0b	000 11b	01b	0Dh	Reservation Register	NVMe Base specification
0b	000 11b	10b	0Eh	Reservation Report	NVMe Base specification
0b	001 00b	01b	11h	Reservation Acquire	NVMe Base specification
0b	001 01b	01b	15h	Reservation Release	NVMe Base specification
0b	001 10b	01b	19h	Copy	4.2.7 and NVMe Base specification
0b	111 10b	01b	79h	Zone Management Send	4.3
0b	111 10b	10b	7Ah	Zone Management Receive	4.4
0b	111 11b	01b	7Dh	Zone Append	4.5

NOTES:

1. Opcodes not listed are reserved.
2. All Zoned Namespace Command Set I/O commands use the Namespace Identifier (NSID) field.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.

### 4.1 Command Specific Status Codes

This specification supports the Command Specific status values described in the NVMe Base specification. Command Specific status values that are specific to the Zoned Namespace Command Set specification are defined in this section.

Figure 21 defines the additional Zoned Namespace Command Set Command Specific status values.

**Figure 21: Status Code – Command Specific Status Values, Zoned Namespace Command Set**

Value	Description	Commands Affected
B8h	Zone Boundary Error	Read <sup>1</sup> , Compare <sup>1</sup> , Verify <sup>1</sup> , Copy <sup>1</sup> , Write, Write Uncorrectable, Write Zeroes, Zone Append
B9h	Zone Is Full	Write, Write Uncorrectable, Write Zeroes, Copy, Zone Append

Value	Description	Commands Affected
BAh	Zone Is Read Only	Write, Write Uncorrectable, Write Zeroes, Copy, Zone Append
BBh	Zone Is Offline	Read, Compare, Verify, Copy, Write, Write Uncorrectable, Write Zeroes, Zone Append
BCh	Zone Invalid Write	Write, Write Uncorrectable, Write Zeroes, Copy
BDh	Too Many Active Zones	Write, Write Uncorrectable, Write Zeroes, Copy, Zone Append, Zone Management Send
BEh	Too Many Open Zones	Write, Write Uncorrectable, Write Zeroes, Copy, Zone Append, Zone Management Send
BFh	Invalid Zone State Transition	Zone Management Send
NOTES:		
1. This command is affected if the Read Across Zone Boundaries bit in the Zoned Namespace Command Set specific Identify Namespace data structure is cleared to '0'.		

## 4.2 NVM Command Set I/O Commands – Zoned Namespace Command Set Specific

### 4.2.1 Write command

The Write command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).

#### 4.2.1.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 22.

**Figure 22: Write – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
B9h	<b>Zone Is Full:</b> The accessed zone is in the ZSF:Full state.
BAh	<b>Zone Is Read Only:</b> The accessed zone is in the ZSRO:Read Only state.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.
BCh	<b>Zone Invalid Write:</b> The write to a zone was not at the write pointer.
BDh	<b>Too Many Active Zones:</b> The controller does not allow additional active zones.
BEh	<b>Too Many Open Zones:</b> The controller does not allow additional open zones.

### 4.2.2 Read command

The Read command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).

#### 4.2.2.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 23.

**Figure 23: Read – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.

### 4.2.3 Write Uncorrectable command

The Write Uncorrectable command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).

#### 4.2.3.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 24.

**Figure 24: Write Uncorrectable – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
B9h	<b>Zone Is Full:</b> The accessed zone is in the ZSF:Full state.
BAh	<b>Zone Is Read Only:</b> The accessed zone is in the ZSRO:Read Only state.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.
BCh	<b>Zone Invalid Write:</b> The write to a zone was not at the write pointer.
BDh	<b>Too Many Active Zones:</b> The controller does not allow additional active zones.
BEh	<b>Too Many Open Zones:</b> The controller does not allow additional open zones.

#### 4.2.4 Compare command

The Compare command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).

##### 4.2.4.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 25.

**Figure 25: Compare – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.

#### 4.2.5 Write Zeroes command

The Write Zeroes command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).

##### 4.2.5.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 26

**Figure 26: Write Zeroes – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
B9h	<b>Zone Is Full:</b> The accessed zone is in the ZSF:Full state.
BAh	<b>Zone Is Read Only:</b> The accessed zone is in the ZSRO:Read Only state.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.
BCh	<b>Zone Invalid Write:</b> The write to a zone was not at the write pointer.
BDh	<b>Too Many Active Zones:</b> The controller does not allow additional active zones.
BEh	<b>Too Many Open Zones:</b> The controller does not allow additional open zones.

#### 4.2.6 Verify command

The Verify command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).



#### 4.2.6.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 27.

**Figure 27: Verify – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.

#### 4.2.7 Copy command

The Copy command operates as defined in the NVMe Base specification, with the additional requirements associated with the zone type of the specified zones that the command operates on (refer to section 2.3).

##### 4.2.7.1 Command Completion

Command Completion is as defined in the NVMe Base specification, with the additional Zoned Namespace Command Set Command Specific status values that are defined in Figure 27.

**Figure 28: Copy – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies a Source Range Entry that contains logical blocks in more than one zone or a destination LBA range that contains logical blocks in more than one zone.
B9h	<b>Zone Is Full:</b> The zone specified by the destination LBA range is in the ZSF:Full state.
BAh	<b>Zone Is Read Only:</b> The zone specified by the destination LBA range is in the ZSRO:Read Only state.
BBh	<b>Zone Is Offline:</b> A zone specified by a Source Range Entry or the zone specified by the destination LBA range is in the ZSO:Offline state.
BCh	<b>Zone Invalid Write:</b> The write to the zone was not at the write pointer.
BDh	<b>Too Many Active Zones:</b> The controller does not allow additional active zones.
BEh	<b>Too Many Open Zones:</b> The controller does not allow additional open zones.

#### 4.3 Zone Management Send command

The Zone Management Send command requests an action on one or more zones. The command uses the Data Pointer, Command Dword 10, Command Dword 11 and Command Dword 13 fields. All other command specific fields are reserved.

**Figure 29: Zone Management Send – Command Dword 10 and Command Dword 11**

Bits	Description
63:00	<b>Starting LBA (SLBA):</b> This field specifies the lowest LBA of the zone on which the Zone Send Action is performed. Command Dword 10 contains bits 31:00 of the SLBA; Command Dword 11 contains bits 63:32 of the SLBA.

**Figure 30: Zone Management Send – Command Dword 13**

Bits	Description
31:09	Reserved
08	<b>Select All:</b> If this bit is set to '1', then the SLBA field shall be ignored. If this bit is cleared to '0', then the SLBA field specifies the lowest logical block of the zone. Refer to section 4.3.1 for specific behavior for each Zone Send Action.

Bits	Description		
07:00	<b>Zone Send Action (ZSA):</b> Defines the zone action to be performed for Zone Management Send.		
	<b>Value</b>	<b>Description</b>	<b>Refer to section</b>
	00h	Reserved	
	01h	<b>Close Zone:</b> Close one or more zones.	4.3.1.1
	02h	<b>Finish Zone:</b> Finish one or more zones.	4.3.1.2
	03h	<b>Open Zone:</b> Open one or more zones.	4.3.1.3
	04h	<b>Reset Zone:</b> Reset one or more zones.	4.3.1.4
	05h	<b>Offline Zone:</b> Offline one or more zones.	4.3.1.5
	06h to 0Fh	Reserved	
	10h	<b>Set Zone Descriptor Extension:</b> Attach Zone Descriptor Extension data to a zone.	4.3.1.6
	11h to FFh	Reserved	

If the command completes successfully, depending on the Zone Send Action field and the current states of the zones specified by the command, then that command may affect zones in various ways, including the following:

- a) the zone state may change;
- b) the Zone Descriptor Extension data may change; or
- c) the Zone Descriptor Extension Valid zone attribute bit may change.

If the controller has multiple outstanding Zone Management Send commands that specify one or more of the same zones, then the results are undefined.

If the zoned namespace containing the specified zone is write protected as described in the Namespace Write Protection section in the NVMe Base specification, then the command shall be aborted with a status code of Namespace is Write Protected.

If there are insufficient available Active Resources or insufficient available Open Resources, the command shall be aborted as described defined in section 2.5, and no zone state transition shall occur.

If the command SLBA field does not specify the starting logical block for a zone in the specified zoned namespace and the Select All bit is cleared to '0', then the command shall be aborted with a status code of Invalid Field in Command.

If the Zone Send Action field specifies Set Zone Descriptor Extension, and the Zone Descriptor Extension Size field value in the Zoned Namespace Command Set specific Identify Namespace data structure is cleared to 0h, then the command shall be aborted with a status code of Invalid Field in Command.

For a shared zoned namespace, the method used by hosts to coordinate Zone Management Send commands is outside the scope of this specification.

### 4.3.1 Zone Send Actions

The Zone Management Send command Zone Send Action field defines what action to perform on one or more zones.

#### 4.3.1.1 Close Zone

If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the:

- a) ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state, then the zone shall be is transitioned to the ZSC:Closed state;
- b) ZSC:Closed state, then no change shall be made to the zone state; and
- c) ZSE:Empty state, the ZSF:Full state, the ZSRO:Read Only state or the ZSO:Offline state, then the command shall be aborted with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the:

- a) ZSIO:Implicitly Opened state; and
- b) ZSEO:Explicitly Opened state,

shall be transitioned to the ZSC:Closed state.

#### **4.3.1.2 Finish Zone**

If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the:

- a) ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, or the ZSC:Closed state, then the zone shall be transitioned to the ZSF:Full state;
- b) ZSF:Full state, then no change shall be made to the zone state; and
- c) ZSRO:Read Only state or the ZSO:Offline state, then the command shall be aborted with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the:

- a) ZSIO:Implicitly Opened state;
- b) ZSEO:Explicitly Opened state; and
- c) ZSC:Closed state,

shall be transitioned to the ZSF:Full state.

#### **4.3.1.3 Open Zone**

If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the:

- a) ZSE:Empty state, the ZSIO:Implicitly Opened state or the ZSC:Closed state, then the zone should be transitioned to the ZSEO:Explicitly Opened state;
- b) ZSEO:Explicitly Opened state, then no change shall be made to the zone state; and
- c) ZSF:Full state, the ZSRO:Read Only state or the ZSO:Offline state, then the command shall be aborted with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the ZSC:Closed state should be transitioned to the ZSEO:Explicitly Opened state.

If there are insufficient available Active Resources or insufficient available Open Resources, the command shall be aborted as described in section 2.5 and no zone state transition shall occur.

#### **4.3.1.4 Reset Zone**

If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the:

- a) ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the specified zone shall be transitioned to the ZSE:Empty state;
- b) ZSE:Empty state, then no change shall be made to the zone state; and
- c) ZSRO:Read Only state, or the ZSO:Offline state, then the command shall be aborted with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the:

- a) ZSIO:Implicitly Opened state;
- b) ZSEO:Explicitly Opened state;
- c) ZSC:Closed state; and
- d) ZSF:Full state,

shall be transitioned to the ZSE:Empty state.

If the command completes successfully, then for each affected zone:

- a) the Write Pointer zone attribute in the Zone Descriptor shall be set to the ZSLBA of the zone; and
- b) the following zone attribute bits in the Zone Descriptor shall be cleared to '0':
  - a) Zone Descriptor Extension Valid;
  - b) Finish Zone Recommended;
  - c) Reset Zone Recommended; and
  - d) Zone Finished by Controller.

If the Variable Zone Capacity bit is set to '1' in the Zone Operation Characteristics field in the Zoned Namespace Command Set specific Identify Namespace data structure (refer to Figure 8), then the controller may change the Zone Capacity field in the Zone Descriptor of each affected zone. If the Zone Capacity field is changed for one or more zones, then the Zone Capacity Changed bit shall be set to '1' in completion queue entry Dword 0 (refer to section 4.3.2).

#### 4.3.1.5 Offline Zone

If the Select All bit in Command Dword 13 is cleared to '0', and the zone specified by the SLBA field is in the:

- a) ZSRO:Read Only state, then the specified zone shall be transitioned to the ZSO:Offline state;
- b) ZSO:Offline state, then no change shall be made to the zone state; and
- c) ZSE:Empty state, the ZSIO:Implicitly Opened state, the ZSEO:Explicitly Opened state, the ZSC:Closed state, or the ZSF:Full state, then the command shall be aborted with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the SLBA field shall be ignored and all zones that are in the ZSRO:Read Only state shall be transitioned to the ZSO:Offline state.

#### 4.3.1.6 Set Zone Descriptor Extension

If the Select All bit in Command Dword 13 is cleared to '0' and the zone specified by the SLBA field is in:

- a) the ZSE:Empty state, then the zone should be transitioned to the ZSC:Closed state; and
- b) any state other than the ZSE:Empty state, then the command shall be aborted with a status code of Invalid Zone State Transition.

If the Select All bit is set to '1', then the command shall be aborted with a status Invalid Field in Command.

If there are insufficient available Active Resources, the command shall be aborted as described in section 2.5 and no zone state transition shall occur.

On successful command completion, the Zone Descriptor Extension of the zone shall be set to the data in the data buffer.

#### 4.3.2 Command Completion

When the command is completed, the controller shall post a completion queue entry to the associated I/O Completion Queue indicating the status for the command. Figure 31 defines the Zone Management Send command specific status values.

**Figure 31: Zone Management Send – Command Specific Status Values**

Value	Description
BDh	<b>Too Many Active Zones:</b> The controller does not allow additional active zones.
BEh	<b>Too Many Open Zones:</b> The controller does not allow additional open zones.
BFh	<b>Invalid Zone State Transition:</b> The request is not a valid zone state transition.

Dword 0 of the completion queue entry indicates if the zone capacity of the specified zone has been changed. The definition of Dword 0 of the completion queue entry is in Figure 32.

**Figure 32: Zone Management Send – Completion Queue Entry Dword 0**

Bits	Description
31:01	Reserved
00	<b>Zone Capacity Changed:</b> This bit indicates if the zone capacity has changed for one or more of the zones specified by the command. If this bit is set to '1', then a zone capacity has changed due to this command. If this bit is cleared to '0', then a zone capacity has not changed due to this command.

If a zone capacity change occurred, then the host may use the Zone Management Receive command as described in section 4.4 to determine what has changed.

#### 4.4 Zone Management Receive command

The Zone Management Receive command returns a data buffer that contains information about zones. That information includes characteristics of the zone, the state of the zone, the capacity of the zone, and other information described in section 4.4.2. The host uses this command to determine the current settings for this information.

If this information changes (e.g., as indicated by the Capacity Changed bit set to '1' in the completion queue entry of a Zone Management Send command, or by a Zone Descriptor Changed event), then the host may use this command to determine the current state of this information (e.g., the current capacity of the zone or the Reset Zone Recommended attribute).

The Zone Management Receive command uses the Data Pointer, Command Dword 10, Command Dword 11, Command Dword 12, and Command Dword 13 fields. All other command specific fields are reserved.

**Figure 33: Zone Management Receive – Data Pointer**

Bits	Description
127:00	<b>Data Pointer (DPTR):</b> This field specifies the location of a data buffer where data is transferred from. Refer to the NVMe Base specification for the definition of this field.

**Figure 34: Zone Management Receive – Command Dword 10 and Command Dword 11**

Bits	Description
63:00	<b>Starting LBA (SLBA):</b> This field specifies an LBA in the lowest numbered zone that the Zone Receive Action operates on. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:32.

**Figure 35: Zone Management Receive – Command Dword 12**

Bits	Description
31:00	<b>Number of Dwords:</b> This field specifies the number of dwords to return. If host software specifies a size larger than what the Zone Receive Action data structure returns, then the controller returns the complete result with undefined results for dwords beyond the end of the data of the Zone Receive Action data structure. This is a 0's based value.

**Figure 36: Zone Management Receive – Command Dword 13**

Bits	Description
31:17	Reserved

Bits	Description																					
16	Zone Receive Action Specific Features:																					
	Zone Receive Action	Description																				
	Report Zones	<b>Partial Report:</b> If this bit is cleared to '0', then the value in the Number of Zones fields (refer to Figure 37 and Figure 38) indicates the number of Zone Descriptors that match the criteria in the Zone Receive Action Specific field.																				
	Extended Report Zones	If this bit is set to '1', then the value in the Number of Zones fields (refer to Figure 37 and Figure 38) indicates the number of fully transferred Zone Descriptors in the data buffer.																				
	All other values	Reserved																				
15:08	Zone Receive Action Specific Field:																					
	Zone Receive Action	Description																				
	Report Zones	<b>Reporting Options:</b> <table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>List all zones.</td></tr><tr><td>1h</td><td>List the zones in the ZSE:Empty state.</td></tr><tr><td>2h</td><td>List the zones in the ZSIO:Implicitly Opened state.</td></tr><tr><td>3h</td><td>List the zones in the ZSEO:Explicitly Opened state.</td></tr><tr><td>4h</td><td>List the zones in the ZSC:Closed state.</td></tr><tr><td>5h</td><td>List the zones in the ZSF:Full state.</td></tr><tr><td>6h</td><td>List the zones in the ZSRO:Read Only state.</td></tr><tr><td>7h</td><td>List the zones in the ZSO:Offline state.</td></tr><tr><td>8h to FFh</td><td>Reserved</td></tr></table>	Value	Description	0h	List all zones.	1h	List the zones in the ZSE:Empty state.	2h	List the zones in the ZSIO:Implicitly Opened state.	3h	List the zones in the ZSEO:Explicitly Opened state.	4h	List the zones in the ZSC:Closed state.	5h	List the zones in the ZSF:Full state.	6h	List the zones in the ZSRO:Read Only state.	7h	List the zones in the ZSO:Offline state.	8h to FFh	Reserved
	Value	Description																				
0h	List all zones.																					
1h	List the zones in the ZSE:Empty state.																					
2h	List the zones in the ZSIO:Implicitly Opened state.																					
3h	List the zones in the ZSEO:Explicitly Opened state.																					
4h	List the zones in the ZSC:Closed state.																					
5h	List the zones in the ZSF:Full state.																					
6h	List the zones in the ZSRO:Read Only state.																					
7h	List the zones in the ZSO:Offline state.																					
8h to FFh	Reserved																					
Extended Report Zones																						
All other values	Reserved																					
07:00	Zone Receive Action (ZRA):																					
	Value	Description																				
	00h	<b>Report Zones:</b> Reports Zone Descriptor entries through the Report Zones data structure (refer to Figure 37).																				
	01h	<b>Extended Report Zones:</b> Reports Zone Descriptor entries through the Extended Report Zones data structure (refer to Figure 38). This value is supported if the zoned namespace is formatted with a non-zero Zone Descriptor Extension Size. Otherwise, the command shall be aborted with a status code of Invalid Field in Command.																				
	02h to FFh	Reserved																				

#### 4.4.1 Zone Receive Actions

The Zone Management Receive command Zone Receive Action field defines what action to perform.

##### 4.4.1.1 Report Zones

The Report Zones action returns the Report Zones data structure (refer to Figure 37). The Zone Descriptors of the Report Zones data structure shall:

- report only Zone Descriptors of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the command;
- match the criteria in the Zone Receive Action Specific field; and
- be sorted in ascending order by the ZSLBA value of each zone.

##### 4.4.1.2 Extended Report Zones

The Extended Report Zones action returns the Extended Report Zones data structure (refer to Figure 38). The Zone Descriptors and Zone Descriptor Extensions of the Extended Report Zones data structure shall:

- report only Zone Descriptors and Zone Descriptor Extensions of zones for which the ZSLBA value is greater than or equal to the ZSLBA value of the zone specified by the SLBA value in the command;
- match the criteria in the Zone Receive Action Specific field; and

- c) be sorted in ascending order by the ZSLBA value of each zone.

## 4.4.2 Zone Management Receive Data Structures

### 4.4.2.1 Report Zones Data Structure

Figure 37 defines the Report Zones Data Structure.

**Figure 37: Report Zones Data Structure**

Bytes	Description
07:00	<b>Number of Zones:</b> If the Partial Report bit (refer to Figure 36) is cleared to '0', then this field indicates the number of zones that match the criteria described in section 4.4.1.1.  If the Partial Report bit is set to '1', then this field indicates the number of zones for which complete Zone Descriptors were transferred to the data buffer.  Refer to section 4.4.1.1 for the content of the data buffer.
63:08	Reserved
127:64	<b>Zone Descriptor 0:</b> Contains the Zone Descriptor for the first zone reported, if any (refer to Figure 39).
191:128	<b>Zone Descriptor 1:</b> Contains the Zone Descriptor for the second zone reported, if any (refer to Figure 39).
...	...
$((n+1)*64)+63:(n+1)*64$	<b>Zone Descriptor n:</b> Contains the Zone Descriptor for the last zone reported, if any (refer to Figure 39).

### 4.4.2.2 Extended Report Zones Data Structure

Figure 38 defines the Extended Report Zones Data Structure.

**Figure 38: Extended Report Zones Data Structure**

Bytes	Description
07:00	<b>Number of Zones:</b> If the Partial Report bit (refer to Figure 36) is cleared to '0', then this field indicates the number of zones that match the criteria described in section 4.4.1.2.  If the Partial Report bit is set to '1', then this field indicates the number of zones for which complete Zone Descriptors and complete Zone Descriptor Extensions were transferred to the data buffer.  Refer to section 4.4.1.2 for the content of the data buffer.
63:08	Reserved
127:64	<b>Zone Descriptor 0:</b> Contains the Zone Descriptor for the first zone reported, if any (Refer to Figure 39).
$(ZDES+127):128$	<b>Zone Descriptor Extension 0:</b> Contains the Zone Descriptor Extension for the first zone reported, if any. If no Zone Descriptor Extension is associated with the zone, then the contents of this field are undefined.
$(ZDES+191):(ZDES+128)$	<b>Zone Descriptor 1:</b> Contains the Zone Descriptor for the second zone reported, if any (refer to Figure 39).
$((2*ZDES)+191):(ZDES+192)$	<b>Zone Descriptor Extension 1:</b> Contains the Zone Descriptor Extension for the second zone reported, if any.
...	...
$((n*ZDES)+((n+2)*64)-1):((n*ZDES)+((n+1)*64))$	<b>Zone Descriptor n:</b> Contains the Zone Descriptor for the last zone reported, if any (refer to Figure 39).
$((((n+1)*ZDES)+((n+2)*64)-1):((n*ZDES)+((n+2)*64))$	<b>Zone Descriptor Extension n:</b> Contains the Zone Descriptor Extension for the last zone reported, if any.
NOTES:	
1. ZDES corresponds to the formatted Zone Descriptor Extension Size field in bytes (i.e., value multiplied by 64) (refer to Figure 9).	

### 4.4.2.3 Zone Descriptor Data Structure

Figure 39 defines the Zone Descriptor data structure.

*Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.*



Figure 39: Zone Descriptor Data Structure

Bytes	Description																					
00	Bits	Description																				
	7:4	Reserved.																				
	3:0	<b>Zone Type (ZT):</b> This field indicates the type of the zone.																				
		<table><tr><th>Value</th><th>Definition</th></tr><tr><td>2h</td><td>Sequential Write Required</td></tr><tr><td>All other values</td><td>Reserved</td></tr></table>	Value	Definition	2h	Sequential Write Required	All other values	Reserved														
Value		Definition																				
2h	Sequential Write Required																					
All other values	Reserved																					
01	Bits	Description																				
	7:4	<b>Zone State (ZS):</b> This field indicates the state of the zone.																				
		<table><tr><th>Value</th><th>Definition</th></tr><tr><td>0h</td><td>Reserved</td></tr><tr><td>1h</td><td>Empty</td></tr><tr><td>2h</td><td>Implicitly Opened</td></tr><tr><td>3h</td><td>Explicitly Opened</td></tr><tr><td>4h</td><td>Closed</td></tr><tr><td>5h to Ch</td><td>Reserved</td></tr><tr><td>Dh</td><td>Read Only</td></tr><tr><td>Eh</td><td>Full</td></tr><tr><td>Fh</td><td>Offline</td></tr></table>	Value	Definition	0h	Reserved	1h	Empty	2h	Implicitly Opened	3h	Explicitly Opened	4h	Closed	5h to Ch	Reserved	Dh	Read Only	Eh	Full	Fh	Offline
		Value	Definition																			
0h		Reserved																				
1h		Empty																				
2h		Implicitly Opened																				
3h		Explicitly Opened																				
4h		Closed																				
5h to Ch		Reserved																				
Dh	Read Only																					
Eh	Full																					
Fh	Offline																					
3:0	Reserved																					
02	<b>Zone Attributes (ZA):</b> Indicates attributes for the Zone:																					
	Bits	Description																				
	7	<b>Zone Descriptor Extension Valid (ZDEV):</b> If this bit is set to '1', then Zone Descriptor Extension data is associated with the zone. If this bit is cleared to '0', then no Zone Descriptor Extension data is associated with the zone. Refer to section 5.1.																				
	6:3	Reserved																				
	2	<b>Reset Zone Recommended (RZR):</b> If this bit is set to '1', then the controller recommends that this zone be reset. Refer to section 5.2.																				
	1	<b>Finish Zone Recommended (FZR):</b> If this bit is set to '1', then the controller recommends that this zone be finished. Refer to section 5.3.																				
0	<b>Zone Finished by Controller (ZFC):</b> If this bit is set to '1', then the controller finished this zone due to a Zone Active Excursion. Refer to section 5.4.																					
07:03	Reserved																					
15:08	<b>Zone Capacity (ZCAP):</b> This field contains the maximum number of logical blocks that are available to be written with user data when the zone is in the ZSE:Empty state. This value shall be less than or equal to the Zone Size field (refer to Figure 9).  If the Variable Zone Capacity bit is cleared to '0' in the Zone Operation Characteristics field in the Zoned Namespace Command Set specific Identify Namespace data structure (refer to Figure 8), then this field is equal to the Zone Size field (refer to Figure 9).  If the Variable Zone Capacity bit is set to '1' in the Zone Operation Characteristics field in the Zoned Namespace Command Set specific Identify Namespace data structure (refer to Figure 8), then the zone capacity may change upon successful completion of a Zone Management Send command specifying the Zone Send Action of Reset Zone (refer to section 4.3.1.4).																					
23:16	<b>Zone Start Logical Block Address (ZSLBA):</b> This field contains the 64-bit address of the lowest logical block for a zone.																					
31:24	<b>Write Pointer (WP):</b> This field is the logical block address where the next write operation for this zone should be issued. Refer to section 2.3.1 for the behavior of the write pointer.																					
63:32	Reserved																					



### 4.4.3 Command Completion

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

### 4.5 Zone Append command

The Zone Append command writes data and metadata, if applicable, to the I/O controller for the zone indicated by the ZSLBA field. The controller assigns the data and metadata, if applicable, to a set of logical blocks within the zone. The lowest LBA of the set of logical blocks written is returned in the completion queue entry (refer to section 4.5.1). The host may also specify protection information to include as part of the operation.

This command uses Command Dword 10, Command Dword 11, Command Dword 12, Command Dword 13, Command Dword 14, and Command Dword 15 fields. If the command uses PRPs for the data transfer, then the Metadata Pointer, PRP Entry 1, and PRP Entry 2 fields are used. If the command uses SGLs for the data transfer, then the Metadata SGL Segment Pointer and SGL Entry 1 fields are used. All other command specific fields are reserved.

Write ordering in the case of multiple outstanding Zone Append commands to a zone is undefined and left to the controller.

If the zone specified by the Zone Append command is not a Sequential Write Required zone, then the command shall be aborted with a status code of Invalid Field in Command.

If the ZSLBA field in the Zone Append command does not specify the lowest logical block for a zone, then the command shall be aborted with a status code of Invalid Field in Command.

The AWUN, NAWUN, NABSN, AWUPF, NAWUPF, NABSPF atomicity parameters apply as defined in the Atomic Operations section in the NVMe Base specification to the Zone Append command.

**Figure 40: Zone Append – Metadata Pointer**

Bits	Description
63:00	<b>Metadata Pointer (MPTR):</b> This field contains the Metadata Pointer, if applicable. Refer to the NVMe Base specification for the definition of this field.

**Figure 41: Zone Append – Data Pointer**

Bits	Description
127:00	<b>Data Pointer (DPTR):</b> This field specifies the location of a data buffer where data is transferred from. Refer to the NVMe Base specification for the definition of this field.

**Figure 42: Zone Append – Command Dword 10 and Command Dword 11**

Bits	Description
63:00	<b>Zone Start Logical Block Address (ZSLBA):</b> This field indicates the 64-bit address of the lowest logical block of the zone in which the data and metadata, if applicable, associated with this command is to be stored. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:32.

**Figure 43: Zone Append – Command Dword 12**

Bits	Description
31	<b>Limited Retry (LR):</b> If set to '1', the controller should apply limited retry efforts. If cleared to '0', the controller should apply all available error recovery means to write the data to the NVM, as defined in the NVMe Base specification.
30	<b>Force Unit Access (FUA):</b> If set to '1', then for data and metadata, if any, associated with logical blocks specified by the Zone Append command, the controller shall write that data and metadata, if any, to non-volatile media before indicating command completion.  There is no implied ordering with other commands. If cleared to '0', then this bit has no effect, as defined in the NVMe Base specification.
29:26	<b>Protection Information Field (PRINFO):</b> Specifies the Protection Information field, as defined in the NVMe Base specification.

Bits	Description
25	<b>Protection Information Remap PIREMAP</b> : This bit determines the contents of the reference tag written to the media (refer 4.5).  For Type 1 protection, the controller shall abort the command with a status code of Invalid Protection Information if this bit is cleared to '0'.  For Type 3 protection, the controller shall abort the command with a status code of Invalid Protection Information if this bit is set to '1'.
24	Reserved
23:20	<b>Directive Type (DTYPE)</b> : Specifies the Directive Type associated with the Directive Specific field (refer to the NVMe Base specification).
19:16	Reserved
15:00	<b>Number of Logical Blocks (NLB)</b> : This field indicates the number of logical blocks to be written. This is a 0's based value.

**Figure 44: Zone Append – Command Dword 13**

Bits	Description
31:16	<b>Directive Specific (DSPEC)</b> : Specifies the Directive Specific value associated with the Directive Type field (refer to the NVMe Base specification).
15:00	Reserved

**Figure 45: Zone Append – Command Dword 14**

Bits	Description
31:00	<b>Initial Logical Block Reference Tag (ILBRT)</b> : This field specifies the Initial Logical Block Reference Tag value. This field is ignored if the zoned namespace is not formatted to use end-to-end protection information. Refer to section 4.5.2.

**Figure 46: Zone Append – Command Dword 15**

Bits	Description
31:16	<b>Logical Block Application Tag Mask (LBATM)</b> : This field specifies the Application Tag Mask value. This field is ignored if the zoned namespace is not formatted to use end-to-end protection information. Refer to the End-to-end Data Protection section in the NVMe Base specification.
15:00	<b>Logical Block Application Tag (LBAT)</b> : This field specifies the Application Tag value. This field is ignored if the zoned namespace is not formatted to use end-to-end protection information. Refer to the End-to-end Data Protection section in the NVMe Base specification.

#### 4.5.1 Protection Information

For the Zone Append command, the actual LBA where data is written by the command is not known to the host until the Zone Append command completes (refer to the ALBA field described in section 4.4.1). The host is not able to provide the actual LBA of the data in the reference tags in the Protection Information at the time the Zone Append command is issued (refer to the End-to-end Data Protection section of the NVMe Base specification). As a result, handling of LBA based reference tags in the Protection Information is handled as described in this section.

Unless otherwise specified, for the protection information received by the controller from the host, the checks performed by the controller shall be performed as described in the Control of Protection Information Checking – PRCHK section of the NVMe Base specification.

For the Reference Tag written to the media, if the PIREMAP bit is:

- a) cleared to '0', then the controller shall write the Reference Tag to the media per the End-to-end Data Protection section of the NVMe Base specification without modification; or
- b) set to '1', then the controller shall write the protection information to media for the first and subsequent LBAs as follows:
  - A) Media Reference Tag[0] = ILBRT + (ALBA - ZSLBA); and
  - B) Media Reference Tag[n+1] = Media Reference Tag[n] + 1

Where: Media Reference Tag [0] is the Reference Tag written to media for the first LBA, and Media Reference Tag [n+1] is the Reference Tag written to media for all other LBAs.

Note: When writing Protection Information using the Zone Append command with the PIREMAP field set to '1', the Reference Tag written to media is not the value transferred from the host to the controller, but contains the value calculated as described in this section.

The controller shall write all protection information received from the host, other than the Reference Tag, to the media without modification.

## 4.5.2 Command Completion

When the command is completed, the controller shall post a completion queue entry to the associated I/O Completion Queue indicating the status for the command. If the command is successfully completed, then the lowest LBA containing the data written by the command is returned in the Assigned LBA (ALBA) field of the completion queue entry. Dword 0 contains bits 31:00 of the ALBA field; Dword 1 contains bits 63:32 of the ALBA field. If the command is not successfully completed, then the contents of the ALBA field are undefined.

Figure 47 defines the Zone Append command specific status values.

**Figure 47: Zone Append – Command Specific Status Values**

Value	Description
B8h	<b>Zone Boundary Error:</b> The command specifies logical blocks in more than one zone.
B9h	<b>Zone Is Full:</b> The accessed zone is in the ZSF:Full state.
BAh	<b>Zone Is Read Only:</b> The accessed zone is in the ZSRO:Read Only state.
BBh	<b>Zone Is Offline:</b> The accessed zone is in the ZSO:Offline state.
BDh	<b>Too Many Active Zones:</b> The controller does not allow additional active zones.
BEh	<b>Too Many Open Zones:</b> The controller does not allow additional open zones.

## 4.6 Controller Architecture

### 4.6.1 I/O Controller

Figure 48 defines Zoned Namespace Command Set specific commands that are mandatory, optional, and prohibited for an I/O controller.

**Figure 48: I/O Controller – Zoned Namespace Command Set Support**

Command	Command Support Requirements <sup>1</sup>
Zone Management Send	M
Zone Management Receive	M
Zone Append	O
NOTES:	
1. O = Optional, M = Mandatory, P = Prohibited	

Figure 49 defines Zoned Namespace Command Set specific log pages that are mandatory, optional, and prohibited for an I/O controller.

**Figure 49: I/O Controller - Zoned Namespace Command Set Specific Log Page Support**

Log Page	Log Page Support Requirements <sup>1</sup>
Changed Zone List	O

**Figure 49: I/O Controller - Zoned Namespace Command Set Specific Log Page Support**

Log Page	Log Page Support Requirements <sup>1</sup>
<p>NOTES:</p> <p>1. O = Optional, M = Mandatory, P = Prohibited</p>	

## 4.7 Reservations

### 4.7.1 Reservation Types

The NVMe Base specification defines a set of reservation types section and the relationship to the commands defined. Figure 50 defines the Zoned Namespace Command Set specific command behavior in the presence of a reservation.

**Figure 50: Command Behavior in the Presence of a Reservation, Zoned Namespace Command Set Specific**

NVMe Command	Write Exclusive Reservation		Exclusive Access Reservation		Write Exclusive Registrants Only or Write Exclusive All Registrants Reservation		Exclusive Access Registrants Only or Exclusive Access All Registrants Reservation	
	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant	Non-Registrant	Registrant
Read Command Group								
Zone Management Receive	A	A	C	C	A	A	C	A
Write Command Group								
Zone Append Zone Management Send	C	C	C	C	C	A	C	A

Key:  
A definition: A=Allowed, command processed normally by the controller  
C definition: C=Conflict, command aborted by the controller with a status code of Reservation Conflict

## 5 Features

### 5.1 Zone Descriptor Extension

The Zone Descriptor Extension feature allows the host software to associate a small amount of data (e.g., UUID) to a zone. The data is attached to a zone as a Zone Descriptor Extension data and can be accessed by issuing a Zone Management Receive command with a Zone Receive Action of Extended Report Zones (refer to section 4.4.1.2) and specifying that zone. The host software may read the Zone Descriptor Extension Valid zone attribute of that zone to identify if the zone has data associated.

A Zone Descriptor Extension is associated with a zone when:

- a) the zone is in the ZSE:Empty state; and
- b) is transitioned to the ZSC:Closed state by the Zone Management Send command with Zone Send Action of Set Zone Descriptor Extension that specifies the zone and the data to associate in the data buffer (refer to section 4.3.1.6).

Upon successful command completion:

- a) the data in the data buffer is associated with the Zone Descriptor Extension of that zone; and
- b) the Zone Descriptor Extension Valid zone attribute bit of that zone is set '1'.

When data has been associated with a Zone Descriptor Extension, the data is associated until:

- a) the zone transitions to the ZSE:Empty state or the ZSO:Offline state; and
- b) the Zone Descriptor Extension Valid zone attribute bit of that zone is cleared to '0'.

### 5.2 Reset Zone Recommended

A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the ZSF:Full state may notify host software to perform a zone reset operation (refer to section 4.3.1.4). If a controller schedules such an internal operation on a zone, the controller may notify the host by:

- a) setting the Reset Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 39); and
- b) generating a Zone Descriptor Changed event for the specific zone.

If the Reset Zone Recommended zone attribute of a specific zone is set to '1', the Reset Recommended Limit field (refer to Figure 10) indicates the number of seconds before the controller intends to perform an internal operation on the specified zone.

As a zone reset operation destroys data in a specific zone, it is optional for the host software to perform a zone reset operation on zones that have the Reset Zone Recommended zone attribute set to '1'. If the host does not perform a zone reset operation on the specific zone, then the internal operation, which may impact performance, may be performed.

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

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

### 5.3 Finish Zone Recommended

A controller that schedules an internal operation (e.g., background operation on the non-volatile media) on a zone that is in the:

- a) ZSIO:Implicitly Opened state;
- b) ZSEO:Explicitly Opened state; or
- c) ZSC:Closed state

may notify host software to either:

*Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.*

- a) perform a zone finish operation (reset to section 4.3.1.2) on that zone; or
- b) initiate write operations to that zone such that the zone transitions to the ZSF:Full state,

so that the controller may cancel the scheduled internal operation. If a controller schedules such an internal operation on a zone, the controller may notify the host by:

- a) setting the Finish Zone Recommended zone attribute of the specific zone to '1' (refer to Figure 39); and
- b) generating a Zone Descriptor Changed event for the specific zone.

If the Finish Zone Recommended zone attribute of a specific zone is set to '1', the Finish Recommended Limit field (refer to Figure 10) indicates the number of seconds before the controller intends to perform an internal operation on the specified zone.

It is optional for the host software to process the above mitigating actions on zones that have the Finish Zone Recommended zone attribute set to '1'. If the host does not perform the mitigating actions on the specific zone, then the internal operation, which may impact performance, may be performed.

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

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

## 5.4 Zone Active Excursions

A Zone Active Excursion is a vendor specific action on a zone that is in the:

- a) ZSIO:Implicitly Opened state;
- b) ZSEO:Explicitly Opened state; or
- c) ZSC:Closed state,

and transitions the zone to the ZSF:Full state. A Zone Active Excursion is orthogonal to the Finish Zone Recommended feature (refer to section 5.3), and can occur at any point when a zone has transitioned to one of the listed zone states. If the controller performs a Zone Active Excursion on such a zone, then the controller shall notify the host by:

- a) setting the zone attribute Zone Finished by Controller bit of that zone to '1' (refer to Figure 39); and
- b) generating a Zone Descriptor Changed event for that zone.

## **Annex A. Zoned Namespaces Host Considerations (Normative)**

### **A.1 Introduction**

The Zoned Namespace Command Set adds host management requirements in addition to the existing NVM Command Set.

To facilitate better interactions between hosts and NVM subsystems, this annex describes the Zoned Namespace feature from a host perspective.

### **A.2 Writing to Zones**

In a Sequential Write Required zone, writes to a zone are required to start at the valid write pointer address.

### **A.3 Open Zone Considerations**

#### **A.3.1 Overview**

Each zone that is in either the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state has both Active Resources and Open Resources attached. Limitations on the availability of Active Resources and Open Resources and the number of zones allowed in these zone states are indicated by the Maximum Open Resources field and the Maximum Active Resources field.

Each zone that is in the ZSC:Closed state has only Active Resources attached. Availability of Active Resources and the number of zones allowed in this zone state are indicated by the Maximum Active Resources field.

If a write operation is processed on LBAs in a zone in the ZSE:Empty state or the ZSC:Closed state, then that zone shall be managed as if that zone is requested to transition to the ZSIO:Implicitly Opened state and shall account for Active Resources and Open Resources as defined in section 2.5.

A zone remains in the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state until:

- a) a write operation is processed for this zone that causes the zone to reach its zone capacity;
- b) the zone state is ZSIO:Implicitly Opened state, and the zone is selected by the controller (which zone is selected is vendor specific) to transition to the ZSC:Closed state;
- c) a Zone Management Send command with Zone Send Action of Finish Zone is issued to the zone;
- d) a Zone Management Send command with Zone Send Action of Reset Zone is issued to the zone;
- e) a Zone Management Send command with Zone Send Action of Close Zone is issued to the zone;
- f) the controller transitions the state of the zone to the ZSRO:Read Only state or the ZSO:Offline state;
- g) the controller transitions the zone to the ZSF:Full state due to a Zone Active Excursion; and
- h) an NVM Subsystem Reset occurs (refer to the Resets section of the NVMe Base specification).

#### **A.3.2 Zones in the ZSEO:Explicitly Opened Zones and the ZSIO:Implicitly Opened states**

An open zone is a zone that is in either the ZSIO:Implicitly Opened state or the ZSEO:Explicitly Opened state. For a zone with zone state ZSE:Empty state or ZSC:Closed state, the controller transitions that zone to:

- a) ZSEO:Explicitly Opened state as a result of processing a Zone Management Send command with Zone Send Action of Open Zone; or
- b) ZSIO:Implicitly Opened state as a result of processing a write operation.

If a controller processes a Zone Management Send command with Zone Send Action of Open Zone on a zone in the ZSC:Closed state, then the controller transitions that zone to the ZSEO:Explicitly Opened state.

For resource management purposes, the controller may transition a zone to the ZSC:Closed state if that zone is in the ZSIO:Implicitly Opened state.

### A.3.3 Opening and Closing Zones

A host may use the following techniques to open a zone for writing and following that to close that zone:

- a) Issue a Zone Management Send command with Zone Send Action of Open Zone to transition a zone to the ZSEO:Explicitly Opened state, and following that the host may issue a Zone Management Send command with Zone Send Action of Close Zone to transition that zone to the ZSC:Closed state; or
- b) Issue a write operation to open a zone, and following that the host may issue a Zone Management Send command with Zone Send Action of Close Zone to transition that zone to the ZSC:Closed state.

The controller may also transition any zone from the ZSIO:Implicitly Opened state to the ZSC:Closed state.

A host may use multiple Zone Management Send commands with Zone Send Action of Close Zone to close more zones than the number necessary to satisfy limitations on the number of zones that may be open.

### A.3.4 Zone Send Action of Finish Zone Considerations

A zone may be opened (e.g., with a Zone Management Send command with Zone Send Action of Open Zone) prior to processing a Zone Management Send command with Zone Send Action of Finish Zone.

Regardless of how a zone is opened before processing a Zone Management Send command with Zone Send Action of Finish Zone, completion of the command transitions the zone to the ZSF:Full state, with the result that Active and Open Resources become available.

## A.4 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 ZRO:Read Only state, the host should perform the following actions:

- 1) Transfer the data from that zone to another location; 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.

## A.5 Capacity and Sizes

There are several data structures involved in size and capacity reporting:

**Figure 51: Size and Capacity Fields**

Data Structure	Field	Bytes	Description
NVM Command Set Specific Identify Namespace	Namespace Size (NSZE)	7:0	(NSZE-1) indicates the highest possible LBA in the namespace
	Namespace Capacity (NCAP)	15:8	The maximum number of allocatable logical blocks in the namespace
	Formatted LBA Size (FLBAS)	26	Index into the list of LBA formats. Refer to the NVMe Base specification.

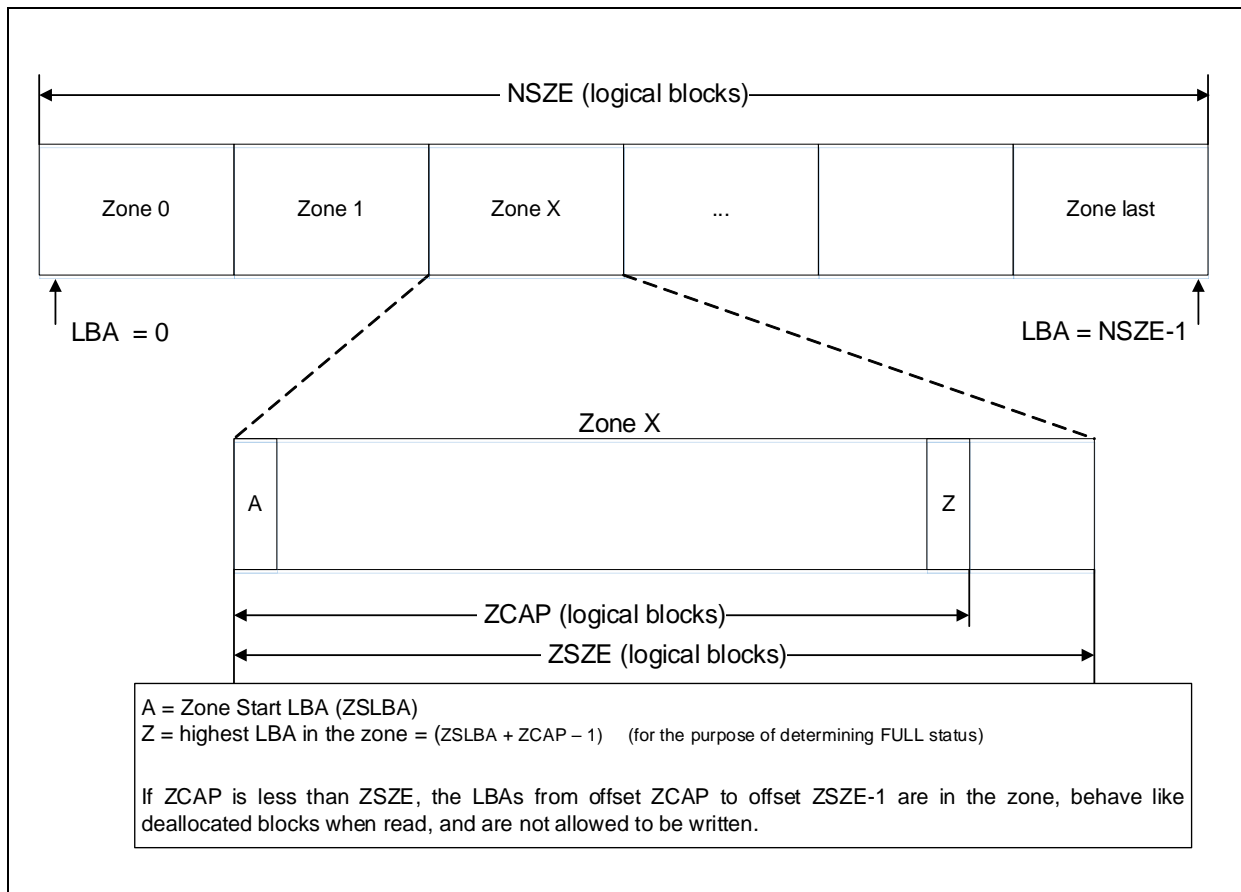


Data Structure	Field	Bytes	Description
	LBA Format XX (LBAFXX)	191:128	LBADS field indicates the logical block data size (in bytes) (as a power of 2). MS field indicates the metadata size (in bytes).
Zoned Namespace Command Set Specific Identify Namespace	LBA Format Extension XX (LBAFEXX)	3071:2816	ZDES indicates the size of the Zone Descriptor Extension (as a multiple of 64 bytes). ZSZE indicates the number of logical blocks of each zone of the namespace.
Zone Descriptor	Zone Capacity (ZCAP)	15:8	Indicates the maximum number of logical blocks that may be written in the zone associated with the data descriptor.

A zone has a size (ZSZE) and a capacity (ZCAP) that are related to optional support for the variable zone capacity feature. ZCAP may change as a result of resetting the write pointer. Refer to Figure 52.

A zone transitions to the ZSF:Full after successful completion of a write operation that includes LBA = (ZSLBA + ZCAP<sub>CURRENT</sub> - 1), in addition to other events (e.g., successful completion of a Zone Management Send command with a Zone Send Action of Finish Zone).

**Figure 52: Zone Size Relationships**



# BASE SPEC CHANGES

## 1 Introduction

...

### 1.9 References

...

<Insert reference to the specification>

NVM Express Zoned Namespace Command Set, Revision 1.0. Available from <https://www.nvmexpress.org>.

...

## 4 Data Structures

...

### 4.6 Common Completion Queue Entry

...

#### 4.6.1 Status Field Definition

...

##### 4.6.1.2 Status Code Type (SCT)

...

##### 4.6.1.2.1 Generic Command Status Definition

**Change Figure 127: Status Code – Generic Command Status Values as shown below** <Refer to the integration of TP 4056b>:

Figure 127: Status Code – Generic Command Status Values

Value	Description	Command Set specific	I/O Command Set(s) <sup>1</sup>
00h	<b>Successful Completion:</b> The command completed without error.	No	
01h	<b>Invalid Command Opcode:</b> A reserved coded value or an unsupported value in the command opcode field.	No	
02h	<b>Invalid Field in Command:</b> A reserved coded value or an unsupported value in a defined field (other than the opcode field). This status code should be used unless another status code is explicitly specified for a particular condition. The field may be in the command parameters as part of the Submission Queue Entry or in data structures pointed to by the command parameters.	No	
03h	<b>Command ID Conflict:</b> The command identifier is already in use. Note: It is implementation specific how many commands are searched for a conflict.	No	
04h	<b>Data Transfer Error:</b> Transferring the data or metadata associated with a command had an error.	No	
05h	<b>Commands Aborted due to Power Loss Notification:</b> Indicates that the command was aborted due to a power loss notification.	No	
06h	<b>Internal Error:</b> The command was not completed successfully due to an internal error. Details on the internal device error should be reported as an	No	

**Figure 127: Status Code – Generic Command Status Values**

Value	Description	Command Set specific	I/O Command Set(s) <sup>1</sup>
	asynchronous event. Refer to Figure 145 for Internal Error Asynchronous Event Information.		
07h	<b>Command Abort Requested:</b> The command was aborted due to an Abort command being received that specified the Submission Queue Identifier and Command Identifier of this command (refer to section 5.1).	No	
08h	<b>Command Aborted due to SQ Deletion:</b> The command was aborted due to a Delete I/O Submission Queue request received for the Submission Queue to which the command was submitted.	No	
09h	<b>Command Aborted due to Failed Fused Command:</b> The command was aborted due to the other command in a fused operation failing.	Yes	N, Z
0Ah	<b>Command Aborted due to Missing Fused Command:</b> The fused command was aborted due to the adjacent submission queue entry not containing a fused command that is the other command in a supported fused operation (refer to section 6.2).	Yes	N, Z
0Bh	<b>Invalid Namespace or Format:</b> The namespace or the format of that namespace is invalid.	No	
0Ch	<b>Command Sequence Error:</b> The command was aborted due to a protocol violation in a multi-command sequence (e.g., a violation of the Security Send and Security Receive sequencing rules in the TCG Storage Synchronous Interface Communications protocol (refer to TCG Storage Architecture Core specification)).	No	
0Dh	<b>Invalid SGL Segment Descriptor:</b> The command includes an invalid SGL Last Segment or SGL Segment descriptor. This may occur when the SGL segment pointed to by an SGL Last Segment descriptor contains an SGL Segment descriptor or an SGL Last Segment descriptor. This may occur when an SGL Last Segment descriptor contains an invalid length (i.e., a length of 0h or 1h that is not a multiple of 16).	No	
0Eh	<b>Invalid Number of SGL Descriptors:</b> There is an SGL Last Segment descriptor or an SGL Segment descriptor in a location other than the last descriptor of a segment based on the length indicated.	No	
0Fh	<b>Data SGL Length Invalid:</b> This may occur if the length of a Data SGL is too short. This may occur if the length of a Data SGL is too long and the controller does not support SGL transfers longer than the amount of data to be transferred as indicated in the SGL Support field of the Identify Controller data structure.	No	
10h	<b>Metadata SGL Length Invalid:</b> This may occur if the length of a Metadata SGL is too short. This may occur if the length of a Metadata SGL is too long and the controller does not support SGL transfers longer than the amount of data to be transferred as indicated in the SGL Support field of the Identify Controller data structure.	No	
11h	<b>SGL Descriptor Type Invalid:</b> The type of an SGL Descriptor is a type that is not supported by the controller.	No	
12h	<b>Invalid Use of Controller Memory Buffer:</b> The attempted use of the Controller Memory Buffer is not supported by the controller. Refer to section 4.7.	No	

**Figure 127: Status Code – Generic Command Status Values**

Value	Description	Command Set specific	I/O Command Set(s) <sup>1</sup>
13h	<b>PRP Offset Invalid:</b> The Offset field for a PRP entry is invalid. This may occur when there is a PRP entry with a non-zero offset after the first entry or when the Offset field in any PRP entry is not dword aligned (i.e., bits 1:0 are not cleared to 00b).	No	
14h	<b>Atomic Write Unit Exceeded:</b> The length specified exceeds the atomic write unit size.	Yes	N, Z
15h	<b>Operation Denied:</b> The command was denied due to lack of access rights. Refer to the appropriate security specification (e.g., TCG Storage Interface Interactions specification). For media access commands, the Access Denied status code should be used instead.	No	
16h	<b>SGL Offset Invalid:</b> The offset specified in a descriptor is invalid. This may occur when using capsules for data transfers in NVMe over Fabrics implementations and an invalid offset in the capsule is specified.	No	
17h	Reserved		
18h	<b>Host Identifier Inconsistent Format:</b> The NVMe subsystem detected the simultaneous use of 64-bit and 128-bit Host Identifier values on different controllers.	No	
19h	<b>Keep Alive Timer Expired:</b> The Keep Alive Timer expired.	No	
1Ah	<b>Keep Alive Timeout Invalid:</b> The Keep Alive Timeout value specified is invalid. This may be due to an attempt to specify a value of 0h on a transport that requires the Keep Alive feature to be enabled. This may be due to the value specified being too large for the associated NVMe Transport as defined in the NVMe Transport binding specification.	No	
1Bh	<b>Command Aborted due to Preempt and Abort:</b> The command was aborted due to a Reservation Acquire command with the Reservation Acquire Action (RACQA) set to 010b (Preempt and Abort).	No	
1Ch	<b>Sanitize Failed:</b> The most recent sanitize operation failed and no recovery action has been successfully completed.	No	
1Dh	<b>Sanitize In Progress:</b> The requested function (e.g., command) is prohibited while a sanitize operation is in progress. Refer to section 8.15.1.	No	
1Eh	<b>SGL Data Block Granularity Invalid:</b> The Address alignment or Length granularity for an SGL Data Block descriptor is invalid. This may occur when a controller supports dword granularity only and the lower two bits of the Address or Length are not cleared to 00b.  Note: An implementation compliant to revision 1.2.1 or earlier may use the status code value of 15h to indicate SGL Data Block Granularity Invalid.	Yes	N, Z
1Fh	<b>Command Not Supported for Queue in CMB:</b> The implementation does not support submission of the command to a Submission Queue in the Controller Memory Buffer or command completion to a Completion Queue in the Controller Memory Buffer.  Note: Revision 1.3 and later of this specification use this status code only for Sanitize commands.	No	
20h	<b>Namespace is Write Protected:</b> The command is prohibited while the namespace is write protected as a result of a change in the namespace write protection state as defined by the Namespace Write Protection State Machine (refer to Figure 489).	No	

**Figure 127: Status Code – Generic Command Status Values**

Value	Description	Command Set specific	I/O Command Set(s) <sup>1</sup>
21h	<b>Command Interrupted:</b> Command processing was interrupted and the controller is unable to successfully complete the command. The host should retry the command.  If this status code is returned, then the controller shall clear the Do Not Retry bit to '0' in the Status field of the CQE (refer to Figure 124). The controller shall not return this status code unless the host has set the Advanced Command Retry Enable (ACRE) field to 1h in the Host Behavior Support feature (refer to section 5.21.1.22).	No	
22h	<b>Transient Transport Error:</b> A transient transport error was detected. If the command is retried on the same controller, the command is likely to succeed. A command that fails with a transient transport error four or more times should be treated as a persistent transport error that is not likely to succeed if retried on the same controller.	No	
23h to 7Fh	Reserved		
80h	<b>LBA Out of Range:</b> The command references an LBA that exceeds the size of the namespace.	Yes	N, Z
81h	<b>Capacity Exceeded:</b> Execution of the command has caused the capacity of the namespace to be exceeded. This error occurs when the Namespace Utilization exceeds the Namespace Capacity, as reported in Figure 245.	No	
82h	<b>Namespace Not Ready:</b> The namespace is not ready to be accessed as a result of a condition other than a condition that is reported as an Asymmetric Namespace Access condition. The Do Not Retry bit indicates whether re-issuing the command at a later time may succeed.	No	
83h	<b>Reservation Conflict:</b> The command was aborted due to a conflict with a reservation held on the accessed namespace. Refer to section 8.8.	No	
84h	<b>Format In Progress:</b> A Format NVM command is in progress on the namespace. The Do Not Retry bit shall be cleared to '0' to indicate that the command may succeed if resubmitted.	No	
85h to BFh	Reserved		
C0h to FFh	Vendor Specific		
Key: N – NVM Command Set Z – Zoned Namespace Command Set NOTES: 1. This column is blank unless the value is I/O Command Set Specific			

...

#### 4.6.1.2.2 Command Specific Status Definition

...

**Change Figure 129: Status Code – Command Specific Status Values as shown below:**

**Figure 129: Status Code – Command Specific Status Values**

Value	Description	Commands Affected
...	...	...
80h	Conflicting Attributes	Dataset Management, Read, Write
81h	Invalid Protection Information	Compare, Read, Verify, Write, Write Zeroes
82h	Attempted Write to Read Only Range	Copy, Dataset Management, Write, Write Uncorrectable, Write Zeroes
83h	Command Size Limit Exceeded	Dataset Management, Copy
84h to B7Fh	Reserved	
B8h to BFh	Refer to the Zoned Namespace Command Set specification	
...	...	...

**Change Figure 131: Status Code – Media and Data Integrity Error Values as shown below**  
 <Refer to the integration of TP 4056b>:

**Figure 131: Status Code – Media and Data Integrity Error Values**

Value	Description	Command Set specific	Command Set(s)
00h to 7Fh	Reserved		
80h	<b>Write Fault:</b> The write data could not be committed to the media.	No	
81h	<b>Unrecovered Read Error:</b> The read data could not be recovered from the media.	No	
82h	<b>End-to-end Guard Check Error:</b> The command was aborted due to an end-to-end guard check failure.	No	
83h	<b>End-to-end Application Tag Check Error:</b> The command was aborted due to an end-to-end application tag check failure.	No	
84h	<b>End-to-end Reference Tag Check Error:</b> The command was aborted due to an end-to-end reference tag check failure.	No	
85h	<b>Compare Failure:</b> The command failed due to a miscompare during a Compare command.	Yes	N, Z
86h	<b>Access Denied:</b> Access to the namespace and/or LBA range is denied due to lack of access rights. Refer to the appropriate security specification (e.g., TCG Storage Interface Interactions specification).	No	
87h	<b>Deallocated or Unwritten Logical Block:</b> The command failed due to an attempt to read from or verify an LBA range containing a deallocated or unwritten logical block.	Yes	N, Z
88h to BFh	Reserved		
C0h to FFh	Vendor Specific		
Key: N – NVM Command Set Z – Zoned Namespace Command Set			

**Change Figure 140: Opcodes for Admin Commands as shown below** <Refer to the integration of TP 4056b>:

**Figure 140: Opcodes for Admin Commands**

Opcode by Field			Combined Opcode <sup>1</sup>	Namespace Identifier Used <sup>2</sup>	Command	Command Set specific	I/O Command Set <sup>8</sup>
(07) Generic Command	(06:02) Function	(01:00) Data Transfer <sup>3</sup>					
0b	000 00b	00b	00h	No	Delete I/O Submission Queue	<u>No</u>	
0b	000 00b	01b	01h	No	Create I/O Submission Queue	<u>No</u>	
0b	000 00b	10b	02h	Yes	Get Log Page	<u>No</u>	
0b	000 01b	00b	04h	No	Delete I/O Completion Queue	<u>No</u>	
0b	000 01b	01b	05h	No	Create I/O Completion Queue	<u>No</u>	
0b	000 01b	10b	06h	NOTE 6	Identify	<u>No</u>	
0b	000 10b	00b	08h	No	Abort	<u>No</u>	
0b	000 10b	01b	09h	Yes	Set Features	<u>No</u>	
0b	000 10b	10b	0Ah	Yes	Get Features	<u>No</u>	
0b	000 11b	00b	0Ch	No	Asynchronous Event Request	<u>No</u>	
0b	000 11b	01b	0Dh	Yes	Namespace Management	<u>No</u>	
0b	001 00b	00b	10h	No	Firmware Commit	<u>No</u>	
0b	001 00b	01b	11h	No	Firmware Image Download	<u>No</u>	
0b	001 01b	00b	14h	Yes	Device Self-test	<u>No</u>	
0b	001 01b	01b	15h	Yes <sup>4</sup>	Namespace Attachment	<u>No</u>	
0b	001 10b	00b	18h	No	Keep Alive	<u>No</u>	
0b	001 10b	01b	19h	Yes <sup>5</sup>	Directive Send	<u>No</u>	
0b	001 10b	10b	1Ah	Yes <sup>5</sup>	Directive Receive	<u>No</u>	
0b	001 11b	00b	1Ch	No	Virtualization Management	<u>No</u>	
0b	001 11b	01b	1Dh	No	NVMe-MI Send	<u>No</u>	
0b	001 11b	10b	1Eh	No	NVMe-MI Receive	<u>No</u>	
0b	111 11b	00b	7Ch	No	Doorbell Buffer Config	<u>No</u>	
0b	111 11b	11b	7Fh	Refer to the NVMe over Fabrics specification.			
1b	000 00b	00b	80h	Yes	Format NVM	Yes	N, <b>Z</b>
1b	000 00b	01b	81h	NOTE 7	Security Send	Yes	N, <b>Z</b>
1b	000 00b	10b	82h	NOTE 7	Security Receive	Yes	N, <b>Z</b>
1b	000 01b	00b	84h	No	Sanitize	Yes	N, <b>Z</b>
1b	000 01b	10b	86h	NOTE 4	Get LBA Status	Yes	N, <b>Z</b>

Figure 140: Opcodes for Admin Commands

Opcode by Field			Combined Opcode <sup>1</sup>	Namespace Identifier Used <sup>2</sup>	Command	Command Set specific	I/O Command Set <sup>8</sup>
(07)	(06:02)	(01:00)					
Generic Command	Function	Data Transfer 3					
<i>Vendor Specific</i>							
1b	n/a	NOTE 3	C0h to FFh		Vendor specific		
Key: N – NVM Command Set Z – Zoned Namespace Command Set  NOTES: 1. Opcodes not listed are reserved. 2. A subset of commands use the Namespace Identifier (NSID) field. If the Namespace Identifier field is used, then the value FFFFFFFFh is supported in this field unless otherwise indicated in footnotes in this figure that a specific command does not support that value or supports that value only under specific conditions. When this field is not used, the field is cleared to 0h as described in Figure 105. 3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional. 4. This command does not support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh. 5. Support for the Namespace Identifier field set to FFFFFFFFh depends on the Directive Operation (refer to section 9). 6. Use of the Namespace Identifier field depends on the CNS value in the Identify Command as described in Figure 244. 7. The use of the Namespace Identifier is Security Protocol specific. 8. This column is blank unless the command is I/O Command Set Specific							

## 5 Admin Command Set

### 5.2 Asynchronous Event Request command

Modify a portion of Figure 148 (Asynchronous Event Information – Notice) as shown below:



**Figure 148: Asynchronous Event Information – Notice**

Value	Description
00h	<p><b>Namespace Attribute Changed:</b> The Identify Namespace data structure (refer to Figure 245) for one or more namespaces, as well as the Namespace List returned when the Identify command is issued with the CNS field set to 02h, have changed. Host software may use this event as an indication that the host should read the Identify Namespace data structures for each namespace to determine what has changed.</p> <p>Alternatively, host software may request the Changed Namespace List (Log Identifier 04h) (refer to section 5.14.1.4) to determine which namespaces in this controller have changed information in the Identify Namespace data structure since the last time the log page was read.</p> <p>A controller shall not send this event if:</p> <ul style="list-style-type: none"> <li>a) Namespace Utilization (refer to Figure 245) has changed, as this is a frequent event that does not require action by the host;</li> <li>b) the ANAGRPID field (refer to Figure 245) has changed; or</li> <li>c) capacity information (i.e., the NUSE field and the NVMCAP field) returned in the Identify Namespace data structure (refer to Figure 245) changed as a result of an ANA state change.</li> </ul> <p>A controller shall only send this event for changes to the Format Progress Indicator field when bits 6:0 of that field transition from a non-zero value to zero, or from a zero value to a non-zero value.</p>
01h	<p><b>Firmware Activation Starting:</b> The controller is starting a firmware activation process during which command processing is paused. Host software may use CSTS.PP to determine when command processing has resumed. To clear this event, host software reads the Firmware Slot Information log page.</p>
02h	<p><b>Telemetry Log Changed:</b> The controller has saved the controller internal state in the Telemetry Controller-Initiated log page and set the Telemetry Controller-Initiated Data Available field to 1h in that log page. To clear this event, the host issues a Get Log Page command with Retain Asynchronous Event bit cleared to '0' for the Telemetry Controller-Initiated log.</p>
03h	<p><b>Asymmetric Namespace Access Change:</b> The Asymmetric Namespace Access information (refer to section 5.14.1.12) related to an ANA Group that contains namespaces attached to this controller has changed (e.g., an ANA state has changed, an ANAGRPID has changed). The current Asymmetric Namespace Access information for attached namespaces is indicated in the Asymmetric Namespace Access log page (refer to section 5.14.1.12). To clear this event, the host issues a Get Log Page with Retain Asynchronous Event bit cleared to '0' for the Asymmetric Namespace Access log.</p> <p>A controller shall not send this event if:</p> <ul style="list-style-type: none"> <li>a) the change is due to the creation of a namespace (refer to section 5.20); or</li> <li>b) the change is due to the deletion of a namespace (refer to section 5.20),</li> </ul> <p>as the Namespace Attribute Changed event is sent for these changes.</p>
04h	<p><b>Predictable Latency Event Aggregate Log Change:</b> Indicates that event pending entries for one or more NVM Sets (refer to section 5.13.1.11) have been added to the Predictable Latency Event Aggregate log.</p>
05h	<p><b>LBA Status Information Alert:</b> The criteria for generating an LBA Status Information Alert Notice event have been met (refer to section 8.22). Information about Potentially Unrecoverable LBAs is available in the LBA Status Information log page (refer to section 5.14.1.14). To clear this event, the host issues a Get Log Page command with Retain Asynchronous Event bit cleared to '0' for the LBA Status Information log.</p>
06h	<p><b>Endurance Group Event Aggregate Log Page Change:</b> Indicates that event entries for one or more Endurance Groups (refer to section 5.4.1.9) have been added to the Predictable Latency Event Aggregate log. To clear this event, the host issues a Get Log Page command with Retain Asynchronous Event bit cleared to '0' for the Endurance Group Event Aggregate log.</p>
07h to EFh	Reserved
EFh	Refer to the Zoned Namespace Command Set specification
F0h to FFh	Refer to the NVMe over Fabrics specification

...

### 5.13 Get Features command

...

Modify a portion of Figure 186 (Get Features – Feature Identifiers) as shown below <Refer to the integration of TP 4056b. >:

Figure 186: Get Features – Feature Identifiers

Description	Section Defining Format of Attributes Returned	I/O Command Set Specific
...	...	...
LBA Range Type	5.21.1.3	Yes <sup>1</sup>
...	...	...
Error Recovery	5.21.1.5	Yes <sup>1</sup>
...	...	...
LBA Status Information Report Interval	5.21.1.21	Yes <sup>1</sup>
...	...	...
Notes: 1. This feature is used by <del>the NVM Command Set</del> I/O command sets that specify logical blocks (e.g., Command Set Identifier 0h or 2h).		

...

### 5.15 Identify command

...

#### 5.15.1 Identify command overview

...

Modify Figure X1: Command Set Identifiers <Refer to the integration of TP 4056b>:

Figure X1: Command Set Identifiers

Namespace Type Value	Description	Reference Section
00h	NVM Command Set	5.15.2.TBDa.1.2
01h	Reserved	
02h	Zoned Namespace Command Set	Zoned Namespace Command Set specification
03h to 2Fh	Reserved	
30h to 3Fh	Vendor specific	
40h to FFh	Reserved	

...

Modify Figure 248: Identify – CNS Values <Refer to the integration of TP 4056b>:

Figure 248: Identify – CNS Values

CNS Value	O/M <sup>1</sup>	Definition	NSID <sup>2</sup>	CNTID <sup>3</sup>	CSI <sup>3a</sup>	Reference Section
		Active Namespace Management				
00h	M	Identify Namespace data structure for the specified NSID or the common namespace capabilities for the NVM Command Set. <sup>6</sup>	Y	N	N	5.15.2
01h	M	Identify Controller data structure for the controller processing the command. <sup>6</sup>	N	N	N	5.15.2.2

**Figure 248: Identify – CNS Values**

CNS Value	O/M <sup>1</sup>	Definition	NSID <sup>2</sup>	CNTID <sup>3</sup>	CSI <sup>3a</sup>	Reference Section
02h	M	Active Namespace ID list.	Y	N	N	5.15.2.3
03h	M	Namespace Identification Descriptor list for the specified NSID.	Y	N	N	5.15.2.4
04h	O	An NVM Set List (refer to Figure 250) is returned to the host for up to 31 NVM Sets. The list contains entries for NVM Set identifiers greater than or equal to the value specified in the NVM Set Identifier (CDW11.NVMSETID) field.	N	N	N	5.15.2.5
05h	M	Identify I/O Command Set specific Namespace data structure for the specified NSID for the I/O Command Set specified in the CSI field. <sup>6</sup>	Y	N	Y	5.15.2.TBDa
06h	M	Identify I/O Command Set Specific Controller data structure for the controller processing the command. <sup>6</sup>	N	N	Y	5.15.2.TBDb
07h	M	Active Namespace ID list associated with the specified I/O Command Set.	Y	N	Y	5.15.2.TBDc
08h to 0Fh		Reserved				
<b>Controller and Namespace Management</b>						
10h	O <sup>4</sup>	Allocated Namespace ID list.	Y	N	N	5.15.2.6
11h	O <sup>4</sup>	Identify Namespace data structure for the specified allocated NSID.	Y	N	N	5.15.2.7
12h	O <sup>4</sup>	Controller List of controllers attached to the specified NSID.	Y	Y	N	5.15.2.8
13h	O <sup>4</sup>	Controller List of controllers that exist in the NVM subsystem.	N	Y	N	5.15.2.9
14h	O <sup>5</sup>	Primary Controller Capabilities data structure for the specified primary controller.	N	Y	N	5.15.2.10
15h	O <sup>5</sup>	Secondary Controller list of controllers associated with the primary controller processing the command.	N	Y	N	5.15.2.11
16h	O	A Namespace Granularity List (refer to Figure 255) is returned to the host for up to sixteen Namespace Granularity Entries.	N	N	N <sup>7</sup>	5.15.2.12
17h	O	A UUID List (refer to Figure 257) is returned to the host.	N	N	N	5.15.2.13
1Ah	O <sup>4</sup>	I/O Command Set specific Allocated Namespace ID list	Y	N	Y	5.15.2.TBDd
1Bh	O <sup>4</sup>	I/O Command Set specific Identify Namespace data structure.	Y	N	Y	5.15.2.TBDe
1Chh	O	I/O Command Set data structure	N	Y	N	5.15.2.TBD
18h to 1Fh		Reserved				

Figure 248: Identify – CNS Values

CNS Value	O/M <sup>1</sup>	Definition	NSID <sup>2</sup>	CNTID <sup>3</sup>	CSI <sup>3a</sup>	Reference Section
Future Definition						
20h to FFh		Reserved				
NOTES: 1. O/M definition: O = Optional, M = Mandatory. 2. The NSID field is used: Y = Yes, N = No. 3. The CDW10.CNTID field is used: Y = Yes, N = No. 3a. The CDW11.CSI field is used: Y = Yes, N = No. 4. Mandatory for controllers that support the Namespace Management capability (refer to section 8.12). 5. Mandatory for controllers that support Virtualization Enhancements (refer to section 8.5). 6. Selection of a UUID may be supported. Refer to section 8.24. 7. Namespace Granularity only applies to namespaces that are associated with <del>the NVM Command Set</del> I/O command sets that specify logical blocks (i.e., Command Set Identifier 0h or 2h).						

...

### 5.15.2 Identify Data Structures

...

**Modify first paragraph of Section 5.15.2.7 (Identify Namespace data structure for an Allocated Namespace ID (CNS 11h) <Refer to the integration of TP 4056b>:**

#### 5.15.2.7 Identify Namespace data structure for an Allocated Namespace ID (CNS 11h)

An Identify Namespace data structure (refer to ~~Figure 245~~section 5.15.2.1) is returned to the host for the namespace specified in the Namespace Identifier (NSID) field if it is an allocated NSID. If the specified namespace is an unallocated NSID then the controller returns a zero filled data structure. ~~If the specified namespace is not associated with the NVM Command Set either the NVM Command Set or the Zoned Namespace Command Set, then the controller shall abort the command with status code Invalid I/O Command Set.~~

If the specified namespace is an invalid NSID then the controller shall fail the command with a status code of Invalid Namespace or Format. If the NSID field is set to FFFFFFFFh then the controller should fail the command with a status code of Invalid Namespace or Format.

...

#### 5.15.2.12 Namespace Granularity List (CNS 16h):

...

**Modify second paragraph of Section 5.15.2.12 (Namespace Granularity List (CNS 16h) <Refer to the integration of TP 4056b>:**

If the controller supports reporting of Namespace Granularity (refer to section 8.12.1), then a Namespace Granularity List (refer to Figure 255) is returned to the host for up to sixteen namespace granularity descriptors (refer to Figure 256).

The controller shall abort the command with status code Invalid Field in Command if the Command Set Identifier is not associated with ~~the NVM Command Set~~ a namespace that is associated with I/O command sets that specify logical blocks (i.e., Command Set Identifier 0h or 2h).

...

#### 5.15.2. **TBDb** Identify I/O Command Set specific Controller data structure (CNS 06h)

...

**Modify Figure **TBDb**: Command Set Vector <Refer to the integration of TP 4056b>:**

Figure **TBD<sub>b</sub>**: I/O Command Set Vector

Bit	Description
63:3	Reserved
2	<b>Zoned Namespace Command Set:</b> This bit is set to '1' if the Zoned Namespace Command Set is selected. This bit is cleared to '0' if the Zoned Namespace Command Set is not selected.
1	Reserved
0	<b>NVM Command Set:</b> This bit is set to '1' if the NVM Command Set is selected. This bit is cleared to '0' if the NVM Command Set is not selected.

...

**Modify Figure 251: Identify – Identify Controller Data Structure, as shown below <Refer to the integration of TP 4056b>:**

Figure 251: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	O/M <sup>1</sup>	Description
...		
95:92	M	<p><b>Optional Asynchronous Events Supported (OAES):</b> This field indicates the optional asynchronous events supported by the controller. A controller shall not send optional asynchronous events before they are enabled by host software.</p> <p>Bits 31:<del>28</del><del>45</del> are reserved.</p> <p>Bit 27 is reserved for the Zoned Namespace Command Set specification.</p> <p>Bits 26:15 are reserved.</p> <p>Bit 14 is set to '1' if the controller supports the Endurance Group Event Aggregate Log Page Change Notices event. If cleared to '0', then the controller does not support the Endurance Group Event Aggregate Log Page Change Notices event.</p> <p>Bit 13 is set to '1' if the controller supports the LBA Status Information Notices event. If cleared to '0', then the controller does not support the LBA Status Information Notices event.</p> <p>Bit 12 is set to '1' if the controller supports the Predictable Latency Event Aggregate Log Change Notices event. If cleared to '0', then the controller does not support the Predictable Latency Event Aggregate Log Change Notices event.</p> <p>Bit 11 is set to '1' if the controller supports sending Asymmetric Namespace Access Change Notices. If cleared to '0', then the controller does not support the Asymmetric Namespace Access Change Notices event.</p> <p>Bit 10 is reserved.</p> <p>Bit 9 is set to '1' if the controller supports the Firmware Activation Notices event. If cleared to '0', then the controller does not support the Firmware Activation Notices event.</p> <p>Bit 8 is set to '1' if the controller supports the Namespace Attribute Notices event and the associated Changed Namespace List log page. If cleared to '0', then the controller does not support the Namespace Attribute Notices event nor the associated Changed Namespace List log page.</p> <p>Bits 7:0 are reserved.</p>
...		
527:526	M	<p><b>Atomic Write Unit Normal (AWUN):</b> This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during normal operation. This field is specified in logical blocks and is a 0's based value.</p> <p>If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUN field in the Identify Namespace data structure. Refer to section 6.4.</p>

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.

Bytes	O/M <sup>1</sup>	Description
		<p>If a write command is submitted with size less than or equal to the AWUN value, the host is guaranteed that the write command is atomic to the NVM with respect to other read or write commands. If a write command is submitted with size greater than the AWUN value, then there is no guarantee of command atomicity. AWUN does not have any applicability to write errors caused by power failure (refer to Atomic Write Unit Power Fail).</p> <p>A value of FFFFh indicates all commands are atomic as this is the largest command size. It is recommended that implementations support a minimum of 128 KiB (appropriately scaled based on LBA size).</p> <p>This field is specific to namespaces that are associated with I/O command sets that specify logical blocks (i.e., Command Set Identifier 0h or 2h), and shall be cleared to 0h for namespaces that are not associated with I/O command sets that specify logical blocks.</p> <p><del>This field is specific to the NVM Command Set and shall be cleared to 0h if the NVM Command Set is not supported.</del></p>
529:528	M	<p><b>Atomic Write Unit Power Fail (AWUPF):</b> This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during a power fail or error condition.</p> <p>If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUPF field in the Identify Namespace data structure. Refer to section 6.4.</p> <p>This field is specified in logical blocks and is a 0's based value. The AWUPF value shall be less than or equal to the AWUN value.</p> <p>If a write command is submitted with size less than or equal to the AWUPF value, the host is guaranteed that the write is atomic to the NVM with respect to other read or write commands. If a write command is submitted that is greater than this size, there is no guarantee of command atomicity. If the write size is less than or equal to the AWUPF value and the write command fails, then subsequent read commands for the associated logical blocks shall return data from the previous successful write command. If a write command is submitted with size greater than the AWUPF value, then there is no guarantee of data returned on subsequent reads of the associated logical blocks.</p> <p>This field is specific to namespaces that are associated with I/O command sets that specify logical blocks, and shall be cleared to 0h for namespaces that are not associated with I/O command sets that specify logical blocks.</p> <p><del>This field is specific to the NVM Command Set and shall be cleared to 0h if the NVM Command Set is not supported.</del></p>
...		
533:532	M	<p><b>Atomic Compare &amp; Write Unit (ACWU):</b> This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format for a Compare and Write fused operation.</p> <p>If a specific namespace guarantees a larger size than is reported in this field, then the Atomic Compare &amp; Write Unit size for that namespace is reported in the NACWU field in the Identify Namespace data structure. Refer to section 6.4.</p> <p>This field shall be supported if the Compare and Write fused command is supported. This field is specified in logical blocks and is a 0's based value. If a Compare and Write is submitted that requests a transfer size larger than this value, then the controller may abort the command with a status code of Invalid Field in Command. If Compare and Write is not a supported fused command, then this field shall be 0h.</p> <p>This field is specific to namespaces that are associated with I/O command sets that specify logical blocks, and shall be cleared to 0h for namespaces that are not associated with I/O command sets that specify logical blocks.</p>

Bytes	O/M <sup>1</sup>	Description
		<del>This field is specific to the NVM Command Set and shall be cleared to 0h if the NVM Command Set is not supported.</del>
...		

...

## 5.21 Set Features command

...

### 5.21.1 Feature Specific Information

...

#### 5.21.1.1 Asynchronous Event Configuration (Feature Identifier 0Bh)

...

**Modify Figure 291: Asynchronous Event Configuration – Command Dword 11, as shown below:**

**Figure 291: Asynchronous Event Configuration – Command Dword 11**

Bit	Description
31:28	Refer to the NVMe over Fabrics specification
<del>27</del>	<del>Zone Descriptor Changed Notices: Refer to the Zoned Namespace Command Set specification.</del>
<del>27</del> 26:15	Reserved
14	<p><b>Endurance Group Event Aggregate Log Change Notices:</b> This bit determines whether an asynchronous event notification is sent to the host when an event entry for an Endurance Group (refer to section 8.17) has been added to the Endurance Group Event Aggregate log (refer to section 5.14.1.15). If this bit is set to '1', then the Endurance Group Event Aggregate Log Change event is sent to the host when this condition occurs. If this bit is cleared to '0', then the controller shall not send the Endurance Group Event Aggregate Log Change event to the host.</p> <p>If Endurance Groups are not supported and this bit is set to '1', then the Set Features command shall be aborted with status set to Invalid Field in Command.</p>
...	

...

## 6 NVM Command Set

...

**Modify Figure 350: Opcodes for NVM Commands <Refer to the integration of TP 4056b>:**

**Figure 350: Opcodes for NVM Commands**

Opcode by Field			Combined Opcode <sup>1</sup>	Command <sup>2</sup>	Reference Section <sup>5</sup>
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer <sup>3</sup>			
0b	000 00b	00b	00h	Flush <sup>4</sup>	N, <del>Z</del>
0b	000 00b	01b	01h	Write	N, <del>Z</del>
0b	000 00b	10b	02h	Read	N, <del>Z</del>
0b	000 01b	00b	04h	Write Uncorrectable	N, <del>Z</del>

Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2021 NVMe™ Corporation.

**Figure 350: Opcodes for NVM Commands**

0b	000 01b	01b	05h	Compare	N, Z
0b	000 10b	00b	08h	Write Zeroes	N, Z
0b	000 10b	01b	09h	Dataset Management	6.7
0b	000 11b	00b	0Ch	Verify	N, Z
0b	000 11b	01b	0Dh	Reservation Register	6.11
0b	000 11b	10b	0Eh	Reservation Report	6.13
0b	001 00b	01b	11h	Reservation Acquire	6.10
0b	001 01b	01b	15h	Reservation Release	6.12
0b	001 10b	01b	19h	Copy	N, Z
0b	111 10b	01b	79h	Zone Management Send	Z
0b	111 10b	10b	7Ah	Zone Management Receive	Z
0b	111 11b	01b	7Dh	Zone Append	Z
<b>Vendor Specific</b>					
1b	n/a	NOTE 3	80h to FFh	Vendor specific	
<p>NOTES:</p> <ol style="list-style-type: none"> <li>1. Opcodes not listed are reserved.</li> <li>2. All NVM commands use the Namespace Identifier (NSID) field. The value FFFFFFFFh is not supported in this field unless footnote 4 in this figure indicates that a specific command does support that value.</li> <li>3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.</li> <li>4. This command may support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.</li> <li>5. Key: N=NVM Command Set Z=Zoned Namespace Command Set (refer to the Zoned Namespace Command Set specification)</li> </ol>					