



#### **LEGAL NOTICE:**

© Copyright 2007 - 2019 NVM Express, Inc. ALL RIGHTS RESERVED.

This NVM Express revision 1.3 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 1.3 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 - 2019 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, Inc.  
3855 SW 153rd Drive  
Beaverton, OR 97003 USA  
info@nvmexpress.org

## NVM Express Technical Proposal for New Feature

<b>Technical Proposal ID</b>	<b>4039a – Supervisory Controller</b>
<b>Change Date</b>	<b>3/4/2019</b>
<b>Builds on Specification</b>	<b>NVM Express 1.3b</b>
<b>References Technical proposals</b>	<b>TP 4003c I/O Determinism TP 4004a ANA Base Protocol TP 4018a NVM Sets and Read Recovery Level TP 4014 Sanitize Enhancements TP 4016 Rebuild Assist TP 4030 Verify Command TP 4033 Enhanced Command Retry TP 4050 Endurance Group Information Enhancements</b>

### Technical Proposal Author(s)

<b>Name</b>	<b>Company</b>
Peter Onufryk	Microsemi
Nick Adams	Intel
Michael Allison	Intel

This TPAR defines an administrative controller. An administrative controller is a controller that supports functionality intended to provide host management capabilities, such as the NVMe-MI in-band mechanism. The primary benefit of an administrative controller in PCIe is that it has a unique Programming Interface allowing a special NVMe driver to be loaded.

### Revision History

<b>Revision Date</b>	<b>Change Description</b>
3/15/2018	<ul style="list-style-type: none"><li>Initial draft</li></ul>
4/11/2018	<ul style="list-style-type: none"><li>Updated controller type in Identify Controller Data Structure field based on workgroup feedback.</li><li>Updated definitions.</li><li>Initial draft of controller architecture changes.</li></ul>
5/9/2018	<ul style="list-style-type: none"><li>Moved supported features, commands, and log pages from supervisory controller section to general spec tables.</li><li>Added entries for discovery controller. NVMe over fabrics specification doesn't discuss features supported by a discovery controller.</li><li>Made editorial updates to the text.</li></ul>
5/13/2018	<ul style="list-style-type: none"><li>Updated text based on workgroup email feedback. The changes were mostly editorial. Changed some table entries from optional to prohibited. Added controller initialization and shutdown sections.</li></ul>

*Technical input submitted to the NVM Express™ Workgroup is subject to the terms of the NVM Express™ Participant's agreement. Copyright © 2014-2019 NVMe™ Corporation.*

5/22/2018	<ul style="list-style-type: none"> <li>• Incorporated feedback from 5/17/2018 workgroup meeting <ul style="list-style-type: none"> <li>◦ Changed Async Event Request from optional to mandatory</li> <li>◦ Changed I/O Command Set Specific Admin commands to prohibited for discovery controller.</li> <li>◦ Removed entries for NVMe over fabrics log pages (leave row blank).</li> <li>◦ I kept get and set features as optional since all of the features are either prohibited or optional. So if an implementation doesn't implement any features, the get/set features command is not needed.</li> </ul> </li> <li>• Reworked supervisory controller text in Section 7.1.3.</li> <li>• Clarified that the Controller List Invalid command status value is returned when a Namespace Attachment command is process that contains a supervisory controller in the controller list.</li> <li>• Updated controller abbreviation footnotes and O/M/P definition footnotes.</li> <li>• Incorporated Jason's feedback.</li> <li>• Made minor editorial updates.</li> </ul>
6/6/2018	<ul style="list-style-type: none"> <li>• Replaced the term supervisory controller with administrative controller.</li> <li>• Moved command, feature, and log page support from generic tables to sections describing controllers (per workgroup request).</li> <li>• Added behavior for a controller that supports no I/O Command Sets</li> <li>• Added descriptive text for the various controller types.</li> </ul>
6/13/2018	<ul style="list-style-type: none"> <li>• Added clarification that administrative controller is require to support Admin command set and may support one or more I/O Command Sets. If an I/O Command Set is supported, then only the I/O Command Set specific Admin commands may be supported.</li> <li>• Updated IOCQES and IOSQES to indicate what is reported when the controller doesn't support I/O queues.</li> <li>• Incorporated reviewer feedback.</li> </ul>
6/20/2018	<ul style="list-style-type: none"> <li>• Incorporated notes to document non-technical changes required to the text in phase 3.</li> </ul>
8/21/2018	<ul style="list-style-type: none"> <li>• Editorial updates to align TP text with workgroup feedback for phase 3 text.</li> </ul>
8/28/2018	<ul style="list-style-type: none"> <li>• Incorporated review feedback.</li> </ul>
9/4/2018	<ul style="list-style-type: none"> <li>• Incorporated review feedback</li> <li>• Added technical writer notes</li> </ul>
9/11/2018	<ul style="list-style-type: none"> <li>• Removed "shall" statements in definitions if the shall statement was in the main part of the specification.</li> </ul>
10/27/2018	<ul style="list-style-type: none"> <li>• Integration</li> </ul>
1/17/2019	<ul style="list-style-type: none"> <li>• Added new commands, Feature Identifiers, and Log Pages from TP 4003a, TP 4004a, TP 4018a, TP 4014, TP 4016, TP 4030, TP 4033, and TP 4050.</li> </ul>
1/22/2019	<ul style="list-style-type: none"> <li>• Fixed the usage of NVM Controller in the Read and Write commands.</li> </ul>
1/23/2019	<ul style="list-style-type: none"> <li>• Adding in technical write updates. Removed underscore in command names in tables.</li> </ul>
1/24/2019	<ul style="list-style-type: none"> <li>• Aligned fonts in a table.</li> </ul>
2/27/2019	<ul style="list-style-type: none"> <li>• Integration</li> </ul>
3/4/2019	<ul style="list-style-type: none"> <li>• Ratified</li> </ul>

**Incompatible Changes:**

- Removed the requirement that at least one I/O command set is supported.

**Notes to technical writer:**

- Please make sure that tables reflect all commands, log pages, and features that have been added in TPs developed in parallel with this TP.

**Description of Specification Changes*****Modify Section 1.2 as shown below:*****1.2 Scope**

The specification defines a register interface for communication with a ~~controller non-volatile memory in an NVM subsystem~~. It also defines ~~a standard command sets for use with the NVM subsystem that may be supported by a controller~~. There are three types of controllers with different capabilities:

- a) I/O controllers;
- b) discovery controllers; and
- c) administrative controllers (refer to section 7.1).

~~In this document the generic term controller is often used instead of enumerating specific controller types when applicable controller types may be determined from the context.~~

***Modify Section 1.6.8 as shown below:*****1.6.8 controller**

~~A PCI Express function that implements NVM Express.~~ A controller is the interface between a host and an NVM subsystem. There are three types of controllers:

- a) I/O controllers;
- b) discovery controllers; and
- c) administrative controllers.

~~A controller executes commands submitted by a host on a Submission Queue and posts a completion on a Completion Queue. All controllers implement one Admin Submission Queue and one Admin Completion Queue. Depending on the controller type, a controller may also implement one or more I/O Submission Queues and I/O Completion Queues. When PCI Express is used as the transport, then a controller is a PCI Express function.~~

***Modify Section 1.6.23 as shown below:*****1.6.23 NVM subsystem**

An NVM subsystem includes one or more controllers, zero or more namespaces, ~~and one or more ports~~. ~~An NVM subsystem may include a non-volatile memory storage medium, and an interface between the controller(s) in the NVM subsystem and non-volatile memory storage medium.~~

***Add the section shown below to Section 1.6 in alphabetical order:*****1.6.x host**

An entity that interfaces to an NVM subsystem through one or more controllers and submits commands to

Submission Queues and retrieves command completions from Completion Queues.

#### **1.6.x host memory**

Memory that may be read and written by both a host and a controller and that is not exposed by a controller (i.e., Controller Memory Buffer or Persistent Memory Region). Host memory may be implemented inside or outside a host (e.g., a memory region exposed by a device that is neither the host nor controller).

#### **1.6.x administrative controller**

A controller that exposes capabilities that allow a host to manage an NVM subsystem. An administrative controller does not implement I/O Queues, provide access to data or metadata associated with logical blocks on a non-volatile memory storage medium, or support namespaces attached to the administrative controller (i.e., there are never any active NSIDs).

#### **1.6.x discovery controller**

A controller that exposes capabilities that allow a host to retrieve a Discovery Log Page. A discovery controller does not implement I/O Queues or provide access to a non-volatile memory storage medium. Refer to the NVMe over Fabrics specification for more information.

#### **1.6.x I/O controller**

A controller that implements I/O queues and is intended to be used to access a non-volatile memory storage medium.

***Modify Section 3.1.1 as shown below***

#### **3.1.1 Offset 00h: CAP – Controller Capabilities**

This register indicates basic capabilities of the controller to host software.

Bit	Type	Reset	Description																		
63:56	RO	0h	Reserved																		
55:52	RO	Impl Spec	<b>Memory Page Size Maximum (MPSMAX):</b> This field indicates the maximum host memory page size that the controller supports. The maximum memory page size is (2 ^ (12 + MPSMAX)). The host shall not configure a memory page size in CC.MPS that is larger than this value.																		
51:48	RO	Impl Spec	<b>Memory Page Size Minimum (MPSMIN):</b> This field indicates the minimum host memory page size that the controller supports. The minimum memory page size is (2 ^ (12 + MPSMIN)). The host shall not configure a memory page size in CC.MPS that is smaller than this value.																		
47:46	RO	0h	Reserved																		
45	RO	Impl Spec	<b>Boot Partition Support (BPS):</b> This field indicates whether the controller supports Boot Partitions. If this field is set to '1', the controller supports Boot Partitions. If this field is cleared to '0', the controller does not support Boot Partitions. Refer to section 8.13.																		
44:37	RO	Impl Spec	<b>Command Sets Supported (CSS):</b> This field indicates the I/O Command Set(s) that the controller supports. <del>A minimum of one command set shall be supported. The field is bit significant.</del> If a bit is set to '1', then the corresponding I/O Command Set is supported. If a bit is cleared to '0', then the corresponding I/O Command Set is not supported. <b>Bit 44 is set if no I/O Command Set is supported.</b> <table><tr><th>Bit</th><th>Definition</th></tr><tr><td>37</td><td>NVM command set</td></tr><tr><td>38</td><td>Reserved</td></tr><tr><td>39</td><td>Reserved</td></tr><tr><td>40</td><td>Reserved</td></tr><tr><td>41</td><td>Reserved</td></tr><tr><td>42</td><td>Reserved</td></tr><tr><td>43</td><td>Reserved</td></tr><tr><td>44</td><td><del>Reserved</del>No I/O Command Set is supported (i.e., only the Admin Command Set is supported)</td></tr></table>	Bit	Definition	37	NVM command set	38	Reserved	39	Reserved	40	Reserved	41	Reserved	42	Reserved	43	Reserved	44	<del>Reserved</del> No I/O Command Set is supported (i.e., only the Admin Command Set is supported)
Bit	Definition																				
37	NVM command set																				
38	Reserved																				
39	Reserved																				
40	Reserved																				
41	Reserved																				
42	Reserved																				
43	Reserved																				
44	<del>Reserved</del> No I/O Command Set is supported (i.e., only the Admin Command Set is supported)																				
36	RO	Impl Spec	<b>NVM Subsystem Reset Supported (NSSRS):</b> This field indicates whether the controller supports the NVM Subsystem Reset feature defined in section 7.3.1. This field is set to '1' if the controller supports the NVM Subsystem Reset feature. This field is cleared to '0' if the controller does not support the NVM Subsystem Reset feature.																		
35:32	RO	Impl Spec	<b>Doorbell Stride (DSTRD):</b> Each Submission Queue and Completion Queue Doorbell register is 32-bits in size. This register indicates the stride between doorbell registers. The stride is specified as (2 ^ (2 + DSTRD)) in bytes. A value of 0h indicates a stride of 4 bytes, where the doorbell registers are packed without reserved space between each register. Refer to section 8.6.																		
31:24	RO	Impl Spec	<b>Timeout (TO):</b> This is the worst case time that host software shall wait for CSTS.RDY to transition from: a) '0' to '1' after CC.EN transitions from '0' to '1'; or b) '1' to '0' after CC.EN transitions from '1' to '0'.  This worst case time may be experienced after events such as an abrupt shutdown or activation of a new firmware image; typical times are expected to be much shorter. This field is in 500 millisecond units.																		
23:19	RO	0h	Reserved																		

**Modify Section 3.1.5 as shown below**

### 3.1.5 Offset 14h: CC – Controller Configuration

This register modifies settings for the controller. Host software shall set the Arbitration Mechanism (CC.AMS), the Memory Page Size (CC.MPS), and the Command Set (CC.CSS) to valid values prior to enabling the

controller by setting CC.EN to '1'. Attempting to create an I/O queue before initializing the I/O Completion Queue Entry Size (CC.IOCQES) and I/O Submission Queue Entry Size (CC.IOSQES) should cause a controller to abort Create I/O Completion Queue or Create I/O Submission Queue commands with a status code of Invalid Queue Size.

Bit	Type	Reset	Description								
31:24	RO	0	Reserved								
23:20	RW / RO	0	<p><b>I/O Completion Queue Entry Size (IOCQES):</b> This field defines the I/O Completion Queue entry size that is used for the selected I/O Command Set. The required and maximum values for this field are specified in the Identify Controller data structure in Figure 109 for each I/O Command Set. The value is in bytes and is specified as a power of two (2^n).</p> <p>If the controller does not support I/O queues, then this field shall be read-only with a value of 0h.</p>								
19:16	RW / RO	0	<p><b>I/O Submission Queue Entry Size (IOSQES):</b> This field defines the I/O Submission Queue entry size that is used for the selected I/O Command Set. The required and maximum values for this field are specified in the Identify Controller data structure in Figure 109 for each I/O Command Set. The value is in bytes and is specified as a power of two (2^n).</p> <p>If the controller does not support I/O queues, then this field shall be read-only with a value of 0h.</p>								
...											
06:04	RW	0h	<p><b>I/O Command Set Selected (CSS):</b> This field specifies the Command Set that is selected <del>for use for the I/O Submission Queues</del>. Host software shall only select a supported I/O Command Set, as indicated in CAP.CSS. This field shall only be changed when the controller is disabled (CC.EN is cleared to '0'). The I/O Command Set selected shall be used for all I/O Submission Queues.</p> <p>If bit 44 is set to '1' in the Command Sets Supported (CSS) field, then the value 111b indicates that only the Admin Command Set is supported and that no I/O Command Set or I/O Command Set Specific Admin commands are supported. When only the Admin Command Set is supported, any command submitted on an I/O Submission Queue and any I/O Command Set Specific Admin command submitted on the Admin Submission Queue is completed with status Invalid Command Opcode. If bit 44 is cleared to '0' in the Command Sets Supported (CSS) field, then the value of 111b is reserved.</p> <table><tr><th>Value</th><th>Definition</th></tr><tr><td>000b</td><td>NVM Command Set</td></tr><tr><td>001b – 1140b</td><td>Reserved</td></tr><tr><td>111b</td><td>Admin Command Set only</td></tr></table>	Value	Definition	000b	NVM Command Set	001b – 1140b	Reserved	111b	Admin Command Set only
Value	Definition										
000b	NVM Command Set										
001b – 1140b	Reserved										
111b	Admin Command Set only										
...											

**Modify Figure 109 Identify – Identify Controller Data Structure as shown below:**

101:100	O	<b>Read Recovery Levels Supported (RRLS):</b> If Read Recovery Levels (RRL) are supported, then this field shall be supported. If a bit is set to 1b, then the corresponding Read Recovery Level is supported. If a bit is cleared to 0b, then the corresponding Read Recovery Level is not supported.																																	
		Bit	Definiton	0	Read Recovery Level 0	1	Read Recovery Level 1	2	Read Recovery Level 2	3	Read Recovery Level 3	4	Read Recovery Level 4 – Default <sup>1</sup>	5	Read Recovery Level 5	6	Read Recovery Level 6	7	Read Recovery Level 7	8	Read Recovery Level 8	9	Read Recovery Level 9	10	Read Recovery Level 10	11	Read Recovery Level 11	12	Read Recovery Level 12	13	Read Recovery Level 13	14	Read Recovery Level 14	15	Read Recovery Level 15 – Fast Fail <sup>1</sup>
		Bit	Definiton																																
		0	Read Recovery Level 0																																
		1	Read Recovery Level 1																																
		2	Read Recovery Level 2																																
		3	Read Recovery Level 3																																
		4	Read Recovery Level 4 – Default <sup>1</sup>																																
		5	Read Recovery Level 5																																
		6	Read Recovery Level 6																																
		7	Read Recovery Level 7																																
		8	Read Recovery Level 8																																
		9	Read Recovery Level 9																																
		10	Read Recovery Level 10																																
		11	Read Recovery Level 11																																
		12	Read Recovery Level 12																																
		13	Read Recovery Level 13																																
		14	Read Recovery Level 14																																
		15	Read Recovery Level 15 – Fast Fail <sup>1</sup>																																
		NOTE: 1. If Read Recovery Levels are supported, then this bit shall be set to ‘1’.																																	
444110:102		Reserved																																	
111	M	<b>Controller Type (CNTRLTYPE):</b> This field specifies the controller type. A value of 0h indicates that the controller type is not reported.																																	
		Implementations compliant to version 1.4 or later of this specification shall report a controller type (i.e., the value 0h is reserved and shall not be used). Implementations compliant to an earlier specification version may report a value of 0h to indicate that a controller type is not reported.																																	
		<table><tr><th>Value</th><th>Controller Type</th></tr><tr><td>0h</td><td>Reserved (controller type not reported)</td></tr><tr><td>1h</td><td>I/O Controller</td></tr><tr><td>2h</td><td>Discovery Controller</td></tr><tr><td>3h</td><td>Administrative Controller</td></tr><tr><td>4h - FFh</td><td>Reserved</td></tr></table>	Value	Controller Type	0h	Reserved (controller type not reported)	1h	I/O Controller	2h	Discovery Controller	3h	Administrative Controller	4h - FFh	Reserved																					
		Value	Controller Type																																
		0h	Reserved (controller type not reported)																																
		1h	I/O Controller																																
		2h	Discovery Controller																																
3h	Administrative Controller																																		
4h - FFh	Reserved																																		
127:112	O	<b>FRU Globally Unique Identifier (FGUID):</b> This field contains a 128-bit value that is globally unique for a given Field Replaceable Unit (FRU). Refer to the NVM Express Management Interface (NVMe-MI) specification for the definition of a FRU. This field remains fixed throughout the life of the FRU. This field shall contain the same value for each controller associated with a given FRU.																																	
		<p>This field uses the EUI-64 based 16-byte designator format. Bytes 122:120 contain the 24-bit Organizationally Unique Identifier (OUI) value assigned by the IEEE Registration Authority. Bytes 127:123 contain an extension identifier assigned by the corresponding organization. Bytes 119:112 contain the vendor specific extension identifier assigned by the corresponding organization. See the IEEE EUI-64 guidelines for more information. This field is big endian (refer to section 7.10).</p> <p>When not implemented, this field contains a value of 0h.</p>																																	



**Modify Chapter 5 as shown below:**

## 5 Admin Command Set

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

The Submission Queue Entry (SQE) structure and the fields that are common to all Admin commands are defined in section 4.2. The Completion Queue Entry (CQE) structure and the fields that are common to all Admin commands are defined in section 4.6. The command specific fields in the SQE and CQE structures (i.e., SQE Command Dwords 10-15 and CQE Dword 0) for the Admin Command Set are defined in this section.

For all Admin commands, Dword 14 and 15 are I/O Command Set specific.

Admin commands should not be impacted by the state of I/O queues (e.g., a full I/O completion queue should not delay or stall the Delete I/O Submission Queue command).

Figure 41 defines Admin commands while Figure 42 defines I/O Command Set Specific Admin commands that are specific to the NVM Command Set (i.e., NVM Command Set Specific Admin commands). Refer to Section 7.1 for mandatory, optional, and prohibited commands for the various controller types.

<Note to editor: remove O/M column in Figures 41 and 42>

**Figure 41: Opcodes for Admin Commands**

Opcode by Field			Combined Opcode <sup>2</sup>	O/M <sup>1</sup>	Namespace Identifier Used <sup>3</sup>	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer <sup>4</sup>				
0b	000 00b	00b	00h	M	No	Delete_IO_Submission_Queue
0b	000 00b	01b	01h	M	No	Create_IO_Submission_Queue
0b	000 00b	10b	02h	M	Yes	Get_Log_Page
0b	000 01b	00b	04h	M	No	Delete_IO_Completion_Queue
0b	000 01b	01b	05h	M	No	Create_IO_Completion_Queue
0b	000 01b	10b	06h	M	Yes	Identify
0b	000 10b	00b	08h	M	No	Abort
0b	000 10b	01b	09h	M	Yes	Set_Features
0b	000 10b	10b	0Ah	M	Yes	Get_Features
0b	000 11b	00b	0Ch	M	No	Asynchronous_Event_Request
0b	000 11b	01b	0Dh	⊖	Yes	Namespace Management
0b	001 00b	00b	10h	⊖	No	Firmware_Commit
0b	001 00b	01b	11h	⊖	No	Firmware_Image_Download
0b	001 01b	00b	14h	⊖	Yes	Device Self-test
0b	001 01b	01b	15h	⊖	Yes <sup>6</sup>	Namespace Attachment
0b	001 10b	00b	18h	NOTE 5	No	Keep Alive
0b	001 10b	01b	19h	⊖	Yes <sup>6</sup>	Directive Send
0b	001 10b	10b	1Ah	⊖	Yes <sup>6</sup>	Directive Receive
0b	001 11b	00b	1Ch	⊖	No	Virtualization Management
0b	001 11b	01b	1Dh	⊖	No	NVMe-MI Send
0b	001 11b	10b	1Eh	⊖	No	NVMe-MI Receive
0b	111 11b	00b	7Ch	⊖	No	Doorbell Buffer Config
0b	111 11b	11b	7Fh	⊖	Refer to the NVMe over Fabrics specification.	

Opcode by Field			Combined Opcode <sup>2</sup>	O/M <sup>1</sup>	Namespace Identifier Used <sup>3</sup>	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer <sup>4</sup>				
I/O Command Set Specific						
1b	na	NOTE 4	80h – BFh	⊖		I/O Command Set specific
Vendor Specific						
1b	na	NOTE 4	C0h – FFh	⊖		Vendor specific

NOTES:

1.

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

2.

Opcodes not listed are reserved.

3.

A subset of commands uses the Namespace Identifier field (CDW1.NSID). If the Namespace Identifier field is used, then the value FFFFFFFFh is supported in this field unless footnote 6 in this figure indicates that a specific command does not support that value. When this field is not used, the field shall be cleared to 0h.

4.

Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.

5.

~~For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory.~~

6.

This command does not support the use of the Namespace Identifier field (CDW1.NSID) set to FFFFFFFFh.

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

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

Opcode (07)	Opcode (06:02)	Opcode (01:00)	Opcode <sup>2</sup>	O/M <sup>1</sup>	Namespace Identifier Used <sup>3</sup>	Command
Generic Command	Function	Data Transfer <sup>4</sup>				
1b	000 00b	00b	80h	⊖	Yes	Format NVM
1b	000 00b	01b	81h	⊖	NOTE 5	Security Send
1b	000 00b	10b	82h	⊖	NOTE 5	Security Receive
1b	000 01b	00b	84h	⊖	No	Sanitize
NOTES:						
1. <del>O/M definition: ⊖ = Optional, M = Mandatory.</del>						
2. Opcodes not listed are reserved.						
3. A subset of commands uses the Namespace Identifier field (CDW1.NSID). If the Namespace Identifier field is used, then unless otherwise specified, the value FFFFFFFFh is supported in this field. When this field is not used, the field shall be cleared to 0h.						
4. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.						
5. The use of the Namespace Identifier is Security Protocol specific.						

**Modify Chapter 5.14.1 as shown below:**

### 5.14.1 Log Specific Information

Figure 90 and Figure 91 define the Log pages that may be retrieved with the Get Log Page command and the scope of the information that is returned in those Log pages. **Refer to section 7.1 for mandatory, optional, and prohibited Log pages for the various controller types.**

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

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

*<Note to editor: remove O/M column in Figures 90 and 91>*

**Figure 90: Get Log Page – Log Page Identifiers**

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

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

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

**Modify Chapter 5.19.1 as shown below:**

### 5.19.1 Command Completion

When the command is completed, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.

Command specific status values associated with the Namespace Attachment command are defined in Figure 119. For failures, the byte offset of the first failing entry is reported in the Command Specific Information field of the Error Information Log Entry. The controller does not process further entries in the Controller List after an error is encountered.

**Figure 119: Namespace Attachment – Command Specific Status Values**

Value	Description
18h	<b>Namespace Already Attached:</b> The controller is already attached to the namespace specified.
19h	<b>Namespace Is Private:</b> The controller is not attached to the namespace. The request to attach the controller could not be completed because the namespace is private and is already attached to one controller.
1Ah	<b>Namespace Not Attached:</b> The controller is not attached to the namespace. The request to detach the controller could not be completed.
1Ch	<b>Controller List Invalid:</b> The controller list provided is invalid or the controller list contains an administrative controller.

**Modify Chapter 5.21.1 as shown below:**

### 5.21.1 Feature Specific Information

Figure 128 defines the Features that may be configured with Set Features and retrieved with Get Features. Figure 129 defines Features that are specific to the NVM Command Set. Refer to section 7.1 for mandatory, optional, and prohibited features for the various controller types. Some Features utilize a memory buffer to configure or return attributes for a Feature, whereas others only utilize a Dword in the command or completion queue entry. Feature values that are not persistent across power cycles and resets are restored to their default values as part of a controller reset operation. The default value for each Feature is vendor specific and set by the manufacturer unless otherwise specified; it is not changeable. For more information on Features, including default, saveable, and current value definitions, refer to section 7.8.

There may be commands in execution when a Feature is changed. The new settings may or may not apply to commands already submitted for execution when the Feature is changed. Any commands submitted to a Submission Queue after a Set Features is successfully completed shall utilize the new settings for the associated Feature. To ensure that a Feature applies to all subsequent commands, commands being processed should be completed prior to issuing the Set Features command.

<Note to editor: remove O/M column in Figures 128 and 129>

Figure 128: Set Features – Feature Identifiers

Feature Identifier	O/M <sup>6</sup>	Persistent Across Power Cycle and Reset <sup>2</sup>	Uses Memory Buffer for Attributes	Feature NameDescription
00h	Reserved			
01h	M	No	No	Arbitration
02h	M	No	No	Power Management
03h	O	Yes	Yes	LBA Range Type
04h	M	No	No	Temperature Threshold
05h	M	No	No	Error Recovery
06h	O	No	No	Volatile Write Cache
07h	M	No	No	Number of Queues
08h	NOTE 5	No	No	Interrupt Coalescing
09h	NOTE 5	No	No	Interrupt Vector Configuration
0Ah	M	No	No	Write Atomicity Normal
0Bh	M	No	No	Asynchronous Event Configuration
0Ch	O	No	Yes	Autonomous Power State Transition
0Dh	O	No <sup>3</sup>	No <sup>4</sup>	Host Memory Buffer
0Eh	O	No	Yes	Timestamp
0Fh	O	No	No	Keep Alive Timer
10h	O	Yes	No	Host Controlled Thermal Management
11h	O	No	No	Non-Operational Power State Config
12h – 77h	Reserved			
78h – 7Fh		Refer to the NVMe Management Interface Specification for definition.		
80h – BFh				Command Set Specific (Reserved)
C0h – FFh				Vendor Specific <sup>1</sup>
<p>NOTES:</p> <ol style="list-style-type: none"> <li>1. The behavior of a controller in response to an inactive namespace ID to a vendor specific Feature Identifier is vendor specific.</li> <li>2. This column is only valid if the feature is not saveable (refer to section 7.8). If the feature is saveable, then this column is not used and any feature may be configured to be saved across power cycles and reset.</li> <li>3. The controller does not save settings for the Host Memory Buffer feature across power states and reset events, however, host software may restore the previous values. Refer to section 7.8.</li> <li>4. The feature does not use a memory buffer for Set Features, but it does use a memory buffer for Get Features. Refer to section 8.9.</li> </ol> <p><del>5. The feature is mandatory for NVMe over PCIe. This feature is not supported for NVMe over Fabrics.</del></p> <p>6. O/M: O = Optional, M = Mandatory.</p>				

**Figure 129: Set Features, NVM Command Set Specific – Feature Identifiers**

Feature Identifier	<del>O/M</del> <sup>4</sup>	Persistent Across Power Cycle and Reset <sup>1</sup>	Uses Memory Buffer for Attributes	<del>Feature Name</del> Description
80h	<del>O</del>	Yes	No	Software Progress Marker
81h	<del>O</del> <sup>2</sup>	No	Yes	Host Identifier
82h	<del>O</del> <sup>3</sup>	No	No	Reservation Notification Mask
83h	<del>O</del> <sup>3</sup>	Yes	No	Reservation Persistence
84h – BFh				Reserved
NOTES: 1. This column is only valid if the feature is not saveable (refer to section 7.8). If the feature is saveable, then this column is not used and any feature may be configured to be saved across power cycles and reset. <del>2. Mandatory if reservations are supported as indicated in the Identify Controller data structure.</del> <del>3. Mandatory if reservations are supported by the namespace as indicated by a non-zero value in the Reservation Capabilities (RESCAP) field in the Identify Namespace data structure.</del> <del>4. O/M: O = Optional, M = Mandatory.</del>				

**Modify Chapter 6 as shown below:**

## 6 NVM Command Set

An NVM subsystem is comprised of some number of controllers, where each controller may access some number of namespaces, where each namespace is comprised of some number of logical blocks. A logical block is the smallest unit of data that may be read or written from the controller. The logical block data size, reported in bytes, is always a power of two. Logical block sizes may be 512 bytes, 1KB, 2KB, 4KB, 8KB, etc. Supported logical block sizes are reported in the Identify Namespace data structure.

The NVM Command Set includes the commands listed in Figure 184. ~~Refer to section 7.1 for mandatory, optional, and prohibited commands for the various controller types.~~ The following subsections describe the definition for each of these commands. Commands shall only be submitted by the host when the controller is ready as indicated in the Controller Status register (CSTS.RDY) and after appropriate I/O Submission Queue(s) and I/O Completion Queue(s) have been created.

The Submission Queue Entry (SQE) structure and the fields that are common to all NVM commands are defined in section 4.2. The Completion Queue Entry (CQE) structure and the fields that are common to all NVM commands are defined in section 4.6. The command specific fields in the SQE and CQE structures (i.e., SQE Command Dwords 10-15 and CQE Dword 0) for the NVM Command Set are defined in this section.

In the case of Compare, Read, Write, and Write Zeroes commands, the host may indicate whether a time limit should be applied to error recovery for the operation by setting the Limited Retry (LR) field in the command. The time limit is specified in the Error Recovery feature, specified in section 5.21.1.5. If the host does not specify a time limit should be applied, then the controller should apply all error recovery means to complete the operation.

~~<Note to editor: remove O/M column in Figures 184>~~

**Figure 184: Opcodes for NVM Commands**

Opcode by Field			Combined Opcode <sup>2</sup>	O/M <sup>1</sup>	Command <sup>3</sup>
(07)	(06:02)	(01:00)			
Standard Command	Function	Data Transfer <sup>5</sup>			
0b	000 00b	00b	00h	M	Flush
0b	000 00b	01b	01h	M	Write
0b	000 00b	10b	02h	M	Read
0b	000 01b	00b	04h	O	Write_Uncorrectable
0b	000 01b	01b	05h	O	Compare
0b	000 10b	00b	08h	O	Write Zeroes
0b	000 10b	01b	09h	O	Dataset_Management
0b	000 11b	01b	0Dh	O <sup>4</sup>	Reservation Register
0b	000 11b	10b	0Eh	O <sup>4</sup>	Reservation Report
0b	001 00b	01b	11h	O <sup>4</sup>	Reservation Acquire
0b	001 01b	01b	15h	O <sup>4</sup>	Reservation Release
<b>Vendor Specific</b>					
1b	na	NOTE 5	80h – FFh	O	Vendor specific

NOTES:

- ~~1. O/M definition: O = Optional, M = Mandatory.~~
- Opcodes not listed are reserved.
- All NVM commands use the Namespace Identifier field (CDW1.NSID).
- ~~4. Mandatory if reservations are supported as indicated in the Identify Controller data structure.~~
- 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.

**Modify a portion of section 6.9 (Read command) as shown below:**

The Read command reads data and metadata, if applicable, from the ~~NVM~~/O controller for the LBAs indicated. The command may specify protection information to be checked as part of the read operation.

**Modify a portion of section 6.14 (Write command) as shown below:**

The Write command writes data and metadata, if applicable, to the ~~NVM~~/O controller for the logical blocks indicated. The host may also specify protection information to include as part of the operation.

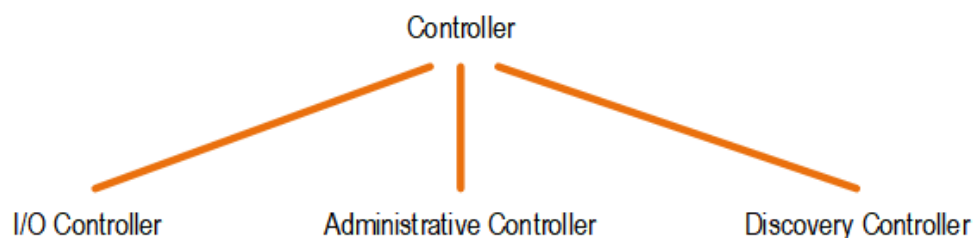
Modify Chapter 7 as shown below:

## 7 Controller Architecture

### 7.1 Introduction

A controller is the interface between a host and an NVM subsystem. As shown in Figure X1, there are three types of controllers. An I/O controller is a general-purpose controller that supports commands that provide access to logical block data and metadata stored on an NVM subsystem's non-volatile storage medium and may support commands that provide management capabilities. An administrative controller is a controller that should support commands that provide management capabilities, but does not support commands that provide access to logical block data and metadata stored on an NVM subsystem's non-volatile storage medium. Finally, a discovery controller is a special type of controller used in NVMe over Fabrics to provide access to a Discovery Log Page.

Figure X1: Controller Types



The Controller Type (CNTRLTYPE) field in the Identify Controller data structure indicates a controller's type. Regardless of controller type, all controllers implement one Admin Submission Queue and one Admin Completion Queue. Depending on the controller type, a controller may also support one or more I/O Submission Queues and I/O Completion Queues.

Host software submits commands to ~~the~~ a controller through pre-allocated Submission Queues. ~~The~~ A controller is alerted to newly submitted commands through SQ Tail Doorbell register writes. The difference between the previous doorbell register value and the current register write indicates the number of commands that were submitted.

~~The~~ A controller fetches ~~the~~ commands from the Submission Queue(s) and ~~transmits them to the NVM subsystem for processing-processes them~~. Except for fused operations, there are no ordering restrictions for processing of ~~the~~ commands within or across Submission Queues. Host software should not ~~place~~ submit commands ~~in the list~~ to a Submission Queue that may not be re-ordered arbitrarily. Data ~~associated with the processing of a command~~ may or may not be committed to the NVM ~~subsystem non-volatile memory storage medium media~~ in the order that commands are ~~received-submitted~~.

Host software submits commands of higher priorities to the appropriate Submission Queues. Priority is associated with the Submission Queue itself, thus the priority of the command is based on the Submission Queue ~~it is issued through to which it was submitted~~. The controller arbitrates across the Submission Queues based on fairness and priority according to the arbitration scheme specified in section 4.13.

Upon completion of commands ~~s execution~~ by the controller the controller presents completion queue entries to the host through the appropriate Completion Queues. If MSI-X or multiple message MSI is in use, then the interrupt vector indicates the Completion Queue(s) with possible new command completions for the host to process. If pin-based interrupts or single message MSI interrupts are used, host software interrogates the Completion Queue(s) for new completion queue entries. The host updates the CQ Head doorbell register to release Completion Queue entries to the controller and clear the associated interrupt.

There are no ordering restrictions for completions to the host. Each completion queue entry identifies the Submission Queue Identifier and Command Identifier of the associated command. Host software uses this



information to correlate the completions with the commands submitted to the Submission Queue(s). Host software is responsible for creating I/O Submission Queues and I/O Completion Queues prior to using those queue pairs to submit commands to the controller. I/O Submission Queues and I/O Completion Queues are created using the Create I/O Submission Queue command (refer to section 5.4) and the Create I/O Completion Queue command (refer to section 5.3).

**Add the following sections as subsections of Section 7.1:**

### 7.1.1 I/O Controller

An I/O controller is a general purpose controller that supports commands that provide access to an NVM subsystem's non-volatile storage medium and may support commands that provide management capabilities.

Figure X2 shows an NVM subsystem with three I/O controllers. I/O controller one has two attached namespaces, private namespace A and shared namespace B. I/O controller two also has two attached namespaces, private namespace C and shared namespace B. I/O controller three has no attached namespaces. At some later point in time shared namespace B may be attached to I/O controller three.

**Figure X2: NVM Subsystem with Three I/O Controllers**

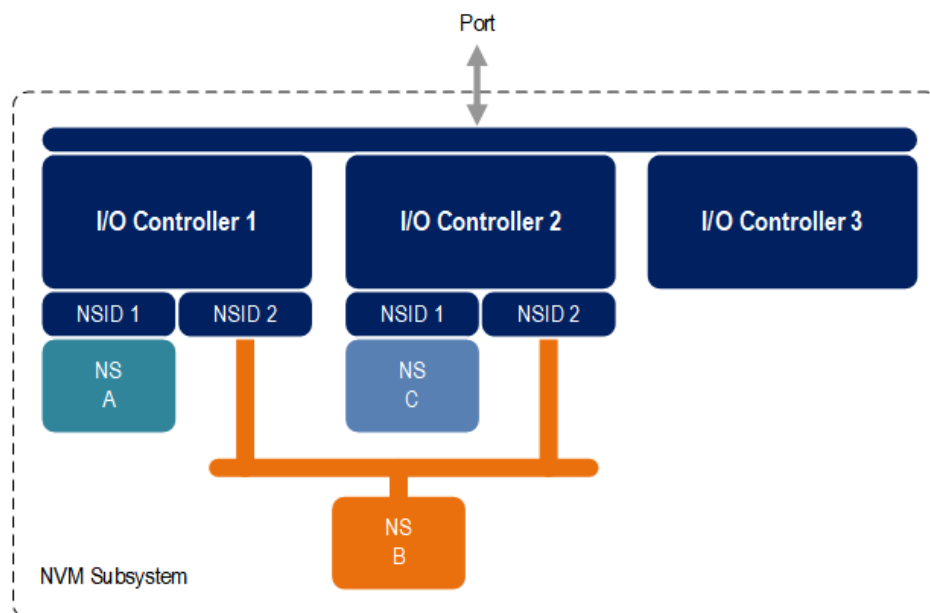


Figure XA1 to Figure XA3 define commands that are mandatory, optional, and prohibited for an I/O controller.

**Figure XA1: I/O Controller – Admin Command Support**

Command	Command Support Requirements <sup>1</sup>
Delete I/O Submission Queue	M
Create I/O Submission Queue	M
Get Log Page	M
Delete I/O Completion Queue	M
Create I/O Completion Queue	M
Identify	M
Abort	M
Set Features	M
Get Features	M
Asynchronous Event Request	M
Namespace Management	O
Firmware Commit	O
Firmware Image Download	O
Device Self-test	O
Namespace Attachment	O
Keep Alive	NOTE 2
Directive Send	O
Directive Receive	O
Virtualization Management	O
NVMe-MI Send	O
NVMe-MI Receive	O
Doorbell Buffer Config	O
NVMe over Fabrics Commands	Refer to the NVMe over Fabrics specification
I/O Command Set Specific Admin Commands	O
Vendor Specific	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory.	

**Figure XA2: I/O Controller – NVM Command Set Specific Admin Command Support**

Command	Command Support Requirements <sup>1</sup>
Format NVM	O
Security Send	O
Security Receive	O
Sanitize	O
Get LBA Status	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited	

**Figure XA3: I/O Controller – NVM Command Set Support**

Command	Command Support Requirements <sup>1</sup>
Flush	M
Write	M
Read	M
Write_Uncorrectable	O
Compare	O
Write Zeroes	O
Dataset_Management	O
Verify	O
Reservation Register	O <sup>2</sup>
Reservation Report	O <sup>2</sup>
Reservation Acquire	O <sup>2</sup>
Reservation Release	O <sup>2</sup>
Vendor Specific	O
Notes:	
1. O = Optional, M = Mandatory, P = Prohibited	
2. Mandatory if reservations are supported as indicated in the Identify Controller data structure.	

Figure XA4 and Figure XA5 define log pages that are mandatory, optional, and prohibited for an I/O controller.

**Figure XA4: I/O Controller – Log Page Support**

Log Page Name	Command Support Requirements <sup>1</sup>
Error Information	M
SMART / Health Information (Controller scope)	M
SMART / Health Information (NVM subsystem scope)	O
Firmware Slot Information	M
Changed Namespace List	O
Commands Supported and Effects	O
Device Self-test	O
Telemetry Host-Initiated	O
Telemetry Controller-Initiated	O
Endurance Group Information	O
Predictable Latency Per NVM Set	O
Predictable Latency Event Aggregate	O
Asymmetric Namespace Access	O
Persistent Event Log	O
LBA Status Information	O
Endurance Group Event Aggregate	O
Notes:	
1. O = Optional, M = Mandatory, P = Prohibited	

**Figure XA5: I/O Controller – NVM Command Set Specific Log Page Support**

Log Page Name	Command Support Requirements <sup>1</sup>
Reservation Notification	O
Sanitize Status	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited	

Figure XA6 and Figure XA7 define features that are mandatory, optional, and prohibited for an I/O controller.

**Figure XA6: I/O Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>
Arbitration	M
Power Management	M
LBA Range Type	O
Temperature Threshold	M
Error Recovery	M
Volatile Write Cache	O
Number of Queues	M
Interrupt Coalescing	NOTE 2
Interrupt Vector Configuration	NOTE 2
Write Atomicity Normal	M
Asynchronous Event Configuration	M
Autonomous Power State Transition	O
Host Memory Buffer	O
Timestamp	O
Keep Alive Timer	O
Host Controlled Thermal Management	O
Non-Operational Power State Config	O
Read Recovery Level Config	O
Predictable Latency Mode Config	O
Predictable Latency Mode Window	O
LBA Status Information Attributes	O
Host Behavior Support	O
Sanitize Config	O
Endurance Group Event Configuration	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. The feature is mandatory for NVMe over PCIe. This feature is not supported for NVMe over Fabrics.	

**Figure XA7: I/O Controller – NVM Command Set Specific Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>
Software Progress Marker	O
Host Identifier	O <sup>2</sup>
Reservation Notification Mask	O <sup>3</sup>
Reservation Persistence	O <sup>3</sup>
Namespace Write Protection Config	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. Mandatory if reservations are supported as indicated in the Identify Controller data structure. 3. Mandatory if reservations are supported by the namespace as indicated by a non-zero value in the Reservation Capabilities (RESCAP) field in the Identify Namespace data structure.	

### 7.1.2 Administrative Controller

An administrative controller is a controller whose intended purpose is to provide NVM subsystem management capabilities. While an I/O controller may support these same management capabilities, an administrative controller has fewer mandatory capabilities. Unlike an I/O controller, an administrative controller does not support commands that provide access to logical block data and metadata stored on an NVM subsystem's non-volatile storage medium. This prevents a host managing an NVM subsystem using an administrative controller from accessing user data. Finally, an administrative controller has a dedicated PCI programming interface value (refer to CC.PI field) allowing a dedicated NVMe management driver to be loaded instead of a generic NVMe driver.

Examples of management capabilities that may be supported by an administrative controller include the following:

- Ability to efficiently poll NVM subsystem health status via NVMe-MI using the NVMe-MI Send and NVMe-MI Receive commands;
- Ability to manage an NVMe enclosure via NVMe-MI using the NVMe-MI Send and NVMe-MI Receive commands;
- Ability to manage NVM subsystem namespaces using the Namespace Attachment and Namespace commands;
- Ability to perform virtualization management using the Virtualization Management command; and
- Ability to reset an entire NVM subsystem using the NVM Subsystem Reset (NSSR) register.

Since an administrative controller does not provide access to logical block data and metadata stored on an NVM subsystem's non-volatile storage medium, it shall not support I/O queues or namespaces attached to the administrative controller.

An administrative controller is required to support the mandatory Admin commands listed in Figure XB1. An administrative controller may support one or more I/O Command Sets. When an Administrative controller supports an I/O Command Set, then only I/O Command Set Specific Admin commands may be supported since an Administrative controller only has an Admin queue and no I/O queues.

Figure X3a shows an NVM subsystem with one administrative controller and two I/O controllers within an NVM subsystem that contains a non-volatile storage medium and namespaces. I/O controller one has two attached namespaces, private namespace A and shared namespace B. I/O controller two also has two attached namespaces, private namespace C and shared namespace B. Since an administrative controller does not provide access to logical block data and metadata stored on an NVM subsystem's non-volatile storage medium, the administrative controller has no attached namespaces. The administrative controller in this example may be used for tasks such as NVM subsystem namespace management and efficiently polling NVM subsystem health status via NVMe-MI. While this example shows a single administrative controller, an NVM subsystem may support zero or more administrative controllers.

**Figure X3a: NVM Subsystem with One Administrative and Two I/O Controllers**

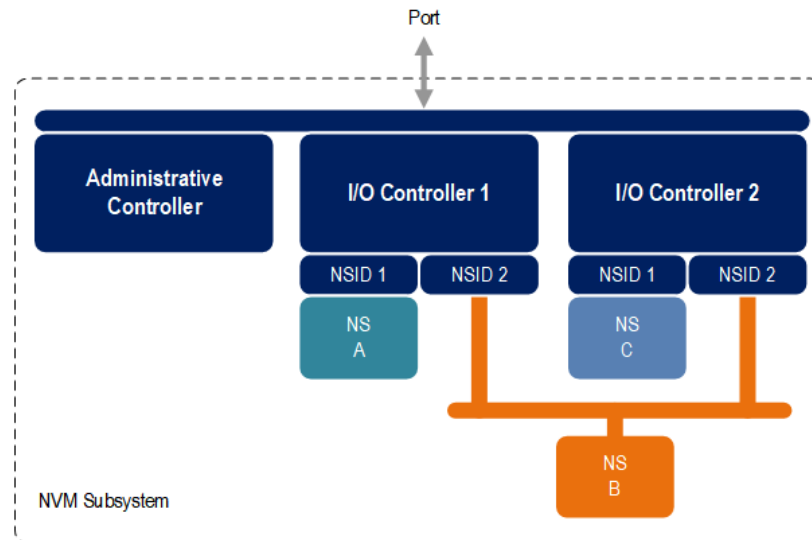


Figure X3b shows an NVM subsystem with one administrative controller within an NVM subsystem that contains no non-volatile storage medium or namespaces. The administrative controller in this example may be used to manage an NVMe enclosure using NVMe-MI. Since the administrative controller is used for a very specific dedicated purpose, the implementer of such an administrative controller may choose to implement only the mandatory capabilities along with the NVMe-MI Send and NVMe-MI Receive commands.

**Figure X3b: NVM Subsystem with One Administrative Controller**

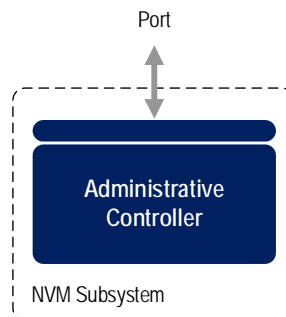


Figure XB1 and Figure XB2 define commands that are mandatory, optional, and prohibited for an administrative controller. Since an administrative controller does not support I/O queues, NVM Command Set commands that are not admin commands are not supported. A host may utilize the Commands Supported and Effects log page to determine optional commands that are supported by an Administrative controller.

**Figure XB1: Administrative Controller – Admin Command Support**

Command	Command Support Requirements <sup>1</sup>
Delete I/O Submission Queue	P
Create I/O Submission Queue	P
Get Log Page	M
Delete I/O Completion Queue	P
Create I/O Completion Queue	P
Identify	M
Abort	O
Set Features	O <sup>3</sup>
Get Features	O <sup>3</sup>
Asynchronous Event Request	O <sup>4</sup>
Namespace Management	O
Firmware Commit	O
Firmware Image Download	O
Device Self-test	O
Namespace Attachment	O
Keep Alive	NOTE 2
Directive Send	O
Directive Receive	O
Virtualization Management	O
NVMe-MI Send	O
NVMe-MI Receive	O
Doorbell Buffer Config	O
NVMe over Fabrics Commands	Refer to the NVMe over Fabrics specification
I/O Command Set Specific Admin Command	O
Vendor Specific	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory. 3. Mandatory if any of the features in Figure XB5 are implemented. 4. Mandatory if Telemetry Log, Firmware Commit or SMART/Health Critical Warnings are supported.	

**Figure XB2: Administrative Controller – NVM Command Set Specific Admin Command Support**

Command	Command Support Requirements <sup>1</sup>
Format NVM	O
Security Send	O
Security Receive	O
Sanitize	O
Get LBA Status	P
Notes: 1. O = Optional, M = Mandatory, P = Prohibited	

Figure XB3 and Figure XB4 defines log pages that are mandatory, optional, and prohibited for an administrative controller.

**Figure XB3: Administrative – Controller Log Page Support**

Log Page Name	Command Support Requirements <sup>1</sup>
Error Information	M
SMART / Health Information (Controller scope)	O
SMART / Health Information (NVM subsystem scope)	O
Firmware Slot Information	O
Changed Namespace List	O
Commands Supported and Effects	M
Device Self-test	O
Telemetry Host-Initiated	O
Telemetry Controller-Initiated	O
Endurance Group Information	O
Predictable Latency Per NVM Set	O
Predictable Latency Event Aggregate	O
Asymmetric Namespace Access	P
Persistent Event Log	O
LBA Status Information	P
Endurance Group Event Aggregate	O
Notes:	
1. O = Optional, M = Mandatory, P = Prohibited	

**Figure XB4: Administrative Controller – NVM Command Set Specific Log Page Support**

Log Page Name	Command Support Requirements <sup>1</sup>
Reservation Notification	P
Sanitize Status	O
Notes:	
1. O = Optional, M = Mandatory, P = Prohibited	

Figure XB5 and Figure XB6 defines features that are mandatory, optional, and prohibited for an administrative controller. If any feature is supported, then the Set Features and Get Features commands shall be supported.



**Figure XB5: Administrative Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>
Arbitration	P
Power Management	O
LBA Range Type	P
Temperature Threshold	O
Error Recovery	P
Volatile Write Cache	P
Number of Queues	P
Interrupt Coalescing	NOTE 2
Interrupt Vector Configuration	NOTE 2
Write Atomicity Normal	P
Asynchronous Event Configuration	O <sup>3</sup>
Autonomous Power State Transition	O
Host Memory Buffer	O
Timestamp	O
Keep Alive Timer	O
Host Controlled Thermal Management	O
Non-Operational Power State Config	O
Read Recovery Level Config	O
Predictable Latency Mode Config	O
Predictable Latency Mode Window	O
LBA Status Information Attributes	P
Host Behavior Support	O
Sanitize Config	O
Endurance Group Event Configuration	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. The feature is optional for NVMe over PCIe. This feature is not supported for NVMe over Fabrics. 3. Mandatory if Telemetry Log, Firmware Commit or SMART/Health Critical Warnings are supported.	

**Figure XB6: Administrative Controller – NVM Command Set Specific Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>
Software Progress Marker	O
Host Identifier	O <sup>2</sup>
Reservation Notification Mask	O <sup>3</sup>
Reservation Persistence	O <sup>3</sup>
Namespace Write Protection Config	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited 2. Mandatory if reservations are supported as indicated in the Identify Controller data structure. 3. Mandatory if reservations are supported by the namespace as indicated by a non-zero value in the Reservation Capabilities (RESCAP) field in the Identify Namespace data structure.	

### 7.1.3 Discovery Controller

A discovery controller is a special type of controller used in NVMe over Fabrics to provide access to a Discovery Log Page. Refer to the NVM Express over Fabrics specification for more information.

**Modify Section 7.6.1 as shown below:**

### **7.6.1 Initialization**

The host should perform the following actions in sequence to initialize the controller to begin executing commands:

1. Set the PCI and PCI Express registers described in section 2 appropriately based on the system configuration. This includes configuration of power management features. A single interrupt (e.g. pin-based, single-MSI, or single MSI-X) should be used until the number of I/O Queues is determined.
2. The host waits for the controller to indicate that any previous reset is complete by waiting for CSTS.RDY to become '0.'
3. The Admin Queue should be configured. The Admin Queue is configured by setting the Admin Queue Attributes (AQA), Admin Submission Queue Base Address (ASQ), and Admin Completion Queue Base Address (ACQ) to appropriate values.
4. The controller settings should be configured. Specifically:
  - a. The arbitration mechanism should be selected in CC.AMS.
  - b. The memory page size should be initialized in CC.MPS.
  - c. The I/O Command Set that is to be used should be selected in CC.CSS or CC.CSS field should be set to the value indicating that only the Admin Command Set is supported.
5. The controller should be enabled by setting CC.EN to '1'.
6. The host should wait for the controller to indicate it is ready to process commands. The controller is ready to process commands when CSTS.RDY is set to '1'.
7. The host should determine the configuration of the controller by issuing the Identify command, specifying the Controller data structure. The host should then determine the configuration of each namespace by issuing the Identify command for each namespace, specifying the Namespace data structure.
8. **If the controller implements I/O queues, then t**~~t~~**he host should determine the number of I/O Submission Queues and I/O Completion Queues supported using the Set Features command with the Number of Queues feature identifier. After determining the number of I/O Queues, the MSI and/or MSI-X registers should be configured.**
9. **If the controller implements I/O queues, then t**~~t~~**he host should allocate the appropriate number of I/O Completion Queues based on the number required for the system configuration and the number supported by the controller. The I/O Completion Queues are allocated using the Create I/O Completion Queue command.**
10. **If the controller implements I/O queues, then t**~~t~~**he host should allocate the appropriate number of I/O Submission Queues based on the number required for the system configuration and the number supported by the controller. The I/O Submission Queues are allocated using the Create I/O Submission Queue command.**
11. If the host desires asynchronous notification of optional events, the host should issue a Set Features command specifying the events to enable. If the host desires asynchronous notification of events, the host should submit an appropriate number of Asynchronous Event Request commands. This step may be done at any point after the controller signals it is ready (i.e., CSTS.RDY is set to '1').

After performing these steps, the controller ~~shall~~**may be ready to process Admin or I/O commands issued by the host.**

For exit of the D3 power state, the initialization steps outlined should be followed. In this case, the number of I/O Submission Queues and I/O Completion Queues shall not change, thus step 7 of the initialization sequence is optional.

**Modify Section 7.6.2 as shown below:**

### **7.6.2 Shutdown**

It is recommended that the host perform an orderly shutdown of the controller by following the procedure in this section when a power-off or shutdown condition is imminent.

The host should perform the following actions in sequence for a normal shutdown:

1. Stop submitting any new I/O commands to the controller and allow any outstanding commands to complete.
2. **If the controller implements I/O queues, then t**~~t~~he host should delete all I/O Submission Queues, using the Delete I/O Submission Queue command. A result of the successful completion of the Delete I/O Submission Queue command is that any remaining commands outstanding are aborted.
3. **If the controller implements I/O queues, then t**~~t~~he host should delete all I/O Completion Queues, using the Delete I/O Completion Queue command.
4. The host should set the Shutdown Notification (CC.SHN) field to 01b to indicate a normal shutdown operation. The controller indicates when shutdown processing is completed by updating the Shutdown Status (CSTS.SHST) field to 10b.