



LEGAL NOTICE:

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

This erratum to the NVM Express revision 1.4 specification is proprietary to the NVM Express, Inc. (also referred to as “Company”) and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this erratum to the NVM Express revision 1.4 specification subject, however, to the Member’s continued compliance with the Company’s Intellectual Property Policy and Bylaws and the Member’s Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: “© **2007 to 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 153rd Drive
Beaverton, OR 97003
USA
info@nvmexpress.org

NVM Express™ Technical Errata

Errata ID	007
Revision Date	2021-03-11
Affected Spec Ver.	NVM Express 1.4b
Corrected Spec Ver.	NVM Express 1.4b

Errata Author(s)

Company	Authors
Brocade	David Peterson
Broadcom	Brad Besmer, James Rizzo
Dell EMC	Austin Bolen, David Black
Intel	Mike Allison, Nick Adams
Kioxia	John Geldman, Paul Suhler
NetApp	Fred Knight
Samsung	Judy Brock
Seagate	Jim Hatfield
Western Digital	Yoni Shternhell

Errata Overview

Misc corrections and clarifications to NVMe 1.4b and NVMe 1.4_NEXT

Revision History

Revision Date	Change Description
2020-09-24	1) Initial version 2) Added changes to figures 167, 398 and 479
2020-09-28	1) Additional changes to figure 167 2) Globally change 'status of' to 'status code'
2020-09-28	1) Added text in section 5.8
2020-09-29	1) Added text in section 5.8
2020-10-05	1) Further changes to 5.8
2020-11-20	1) Added a few new suggestions 2) Reordered by section number 3) Added headings for easier navigation
2020-12-02	1) Added change about CTS Controller status from Kioxia
2020-12-08	1) Added changes for sections 3.1, 4.6.1.2.1, 5.14.1.13.1.6, 5.24.1, and 7.12.2. 2) Corrected the proposed change in section 5.8
2020-12-10	1) Minor tweak to figure 128 2) Added proposed change to section 5.2
2020-12-16	1) Added proposed changes to sections: global consistency issues, 5.1, 5.15.2.2, 5.21.1.14, 7.6.2
2020-12-17	1) Changes to sections 4.6.1.2.1, and 5.1
2021-01-13	1) Added some new discussion topics (at the end of the file) from Judy Brock, Mike Allison, Paul Suhler 2) Added some global discussion topics at the beginning
2021-01-14	1) Added changes to sections: 5.14.1.13.1.12, 8.3.1.5, 8.5, 8.5.1, and 8.20.1 2) Deferred some global changes to a future ECN: when/if, 'status code/status of', adding 'field' or 'bit' after some names. 3) Changed copyright date to 2021 4) Updated the lists of incompatible changes, and the summary of changes
2021-01-27	1) Updated the incompatible Changes section for the change to section 8.15.1. 2) Removed the change to figure 79 as it conflicts with TP 4082. 3) Added the reference to the SGLS field in MPTR field. 4) Removed hyphen in qword-aligned for consistency. 5) Fixed reference to NVMe over Fabrics specification 6) Reformatted text in section 5.8. 7) Added a reference to CDW0.PSDDT in SGL Support field. 8) Changed "timestamp" to "Timestamp field" 9) Clarified that any outstanding command can be aborted as opposed to new submitted commands in section 7.6.2. 10) In section 8.3.1.5 moved Type 3 text to its own bullet for clarity. 11) Added semicolon in new added list item in section 8.5 12) Clarified the pairing of queues in section 8.5.1. 13) Modified figure 479 to use active NSIDs for the Device self-test command as inactive NSIDs are not allowed by that command. 14) Clarified the requirement Sanitize Capabilities field in section 8.15
2021-01-28	1) Accepted changes agreed to in WG meeting 1/28/2021
2021-02-04	2) Final changes to 8.3.1.5 and 8.15.1
2021-03-09	1) Integrated into the NVMe Base Specification
2021-03-11	1) Accepted all changes, removed all comments, and converted references/cross-references to text.

Incompatible Changes

- Section 8.15.1 clarified that not all I/O commands shall be aborted when a sanitize operation is in progress as this conflicts with the Flush command. Section 8.3.1.5 specifies changes to the Flush command.

Summary of changes:

ECN 007: This errata includes:

- Miscellaneous editorial font, capitalization, spelling, references, punctuation, figure title changes, and use of standard phrases
- Clarifications about alignment of metadata pointer when using SGL
- Clarifications about some SGL errors
- Clarification about where to find the Asynchronous Event Type field in the CQE
- Clarification that a Telemetry Log Create Event does not have to be linked to an NVM subsystem fault
- Clarification about the SGSL (SGL Support) field and metadata buffer alignment requirements
- Clarification about timestamp value after a Controller Level Reset
- Add some new status codes that may be returned by the Sanitize command
- Clarify the interaction of the Flush command with the Sanitize command
- Change a table column title and remove an extra space
- Change 'controller data structure' to 'Identify Controller data structure'
- Change 'Identify namespace' to 'identify controller' in a few places, as appropriate
- Clarify what happens to a command issued while controller shutdown is in process
- Corrections to what NVMe 1.4 ECN003 changed regarding PRCHK
- Change some wording to clarify that queue pairs are being referenced in Virtualization Enhancements
- Change 'valid NSID' to 'allocated NSID' in figure 479
- Add an implied requirement that all controllers shall support the same types of sanitization operation types.
- Allow Fabric Disconnect command during Sanitize
- change 'NVMe Subsystem' to 'NVM Subsystem' in a few figures

Description of Specification Changes

Markup Conventions:

Black: Unchanged (however, hot links are removed, references may have changed, but are not part of this ECN: ignore these)

~~Red Strikethrough~~: Deleted

Blue underscore: New

Red Highlighted: TBD values, anchors, and links to be inserted/changed.

<Green Bracketed>: Notes to editor

Specific changes

4.2 Submission Queue Entry – Command Format

<Modify figure 106: command Format – admin and NVM Command set>

...

23:16	<p>Metadata Pointer (MPTR): This field is valid only if the command has metadata that is not interleaved with the logical block data, as specified in the Format NVM command. This is a reserved field in NVMe over Fabrics implementations.</p> <p>If CDW0.PSDT (refer to Figure 105) is cleared to 00b, then this field shall contain the address of a contiguous physical buffer of metadata and that address shall be dword aligned (i.e., bits 1:0 cleared to 00b). The controller is not required to check that bits 1:0 are cleared to 00b. The controller may report an error of Invalid Field in Command if bits 1:0 are not cleared to 00b. If the controller does not report an error of Invalid Field in Command, then the controller shall operate as if bits 1:0 are cleared to 00b.</p> <p>If CDW0.PSDT is set to 01b, then this field shall contain the address of a contiguous physical buffer of metadata and that address may be aligned on any byte boundary. Refer to bit 17 of the SGLS field in the Identify Controller data structure for alignment requirements.</p> <p>If CDW0.PSDT is set to 10b, then this field shall contain the address of an SGL segment that contains exactly one SGL Descriptor. The address of that SGL segment shall be qword aligned (i.e., bits 2:0 cleared to 000b). The SGL Descriptor contained in that SGL segment is the first SGL Descriptor of the metadata for the command. If the SGL Descriptor contained in that SGL segment is an SGL Data Block descriptor, then that SGL Data Block Descriptor is the only SGL Descriptor and therefore describes the entire metadata data transfer. Refer to section 4.4. The controller is not required to check that bits 2:0 are cleared to 000b. The controller may report an error of Invalid Field in Command if bits 2:0 are not cleared to 000b. If the controller does not report an error of Invalid Field in Command, then the controller shall operate as if bits 2:0 are cleared to 000b.</p>
-------	---

4.6.1.2.1 Generic Command Status Definition

<change the description for value 0Dh from a paragraph to a list; change description for status code 0Eh>

Figure 128: Status Code – Generic Command Status Values

Value	Description
...	...
0Dh	<p>Invalid SGL Segment Descriptor: The command includes an invalid SGL Last Segment or SGL Segment descriptor. This may occur when under various conditions, including:</p> <ul style="list-style-type: none">a) 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 whenb) an SGL Last Segment descriptor contains an invalid length (i.e., a length of 0h or 1h that is not a multiple of 16); orc) an SGL Segment descriptor or an SGL Last Segment descriptor contains an invalid address (e.g., an address that is not qword aligned).
0Eh	<p>Invalid Number of SGL Descriptors: 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. This is also used for invalid SGLs in a command capsule (refer to NVMe over Fabrics specification).</p>
...	...

5.2 Asynchronous Event Request command

<clarify where the event type information is found>

...

Asynchronous events are grouped into event types. The event type ~~information~~ is indicated in [the Asynchronous Event Type field in](#) Dword 0 of the completion queue entry for the Asynchronous Event Request command. When the controller posts a completion queue entry for an outstanding Asynchronous Event Request command and thus reports an asynchronous event, subsequent events of that event type are automatically masked by the controller until the host clears that event. An event is cleared by reading the log page associated with that event using the Get Log Page command (refer to section 5.14).

...

<clarify that 'type' means 'event type'>

Figure 146: Asynchronous Event Request – Completion Queue Entry Dword 0

Bits	Description
31:24	Reserved
23:16	Log Page Identifier: Indicates the log page associated with the asynchronous event. This log page needs to be read by the host to clear the event.
15:08	Asynchronous Event Information: Refer to Figure 147, Figure 148, Figure 149, and Figure 150 for detailed information regarding the asynchronous event.
07:03	Reserved
02:00	Asynchronous Event Type: Indicates the event type of the asynchronous event. More specific information on the event is provided in the Asynchronous Event Information field.

5.14.1.13.1.6 Change Namespace Event (Event Type 06h)

<modify figure 226 Change Namespace Event Data Format (Event type 06h) >

...

47:44	Namespace ID (NSID): For a create operation, contains the NSID for the namespace that was created. For a delete operation, contains the NSID from CDW1.NSID (i.e., the NSID for the namespace being deleted or FFFFFFFFh for a delete operation specifying all namespaces).
-------	---

5.14.1.13.1.12 Telemetry Log Create Event (Event Type 0Ch)

<A host is allowed to generate a host-initiated log at any time even when a NVM subsystem fault does not exist>

A Telemetry Log Create event may be created if the controller determines that a host-initiated telemetry log (refer to section 5.15.1.7) or that a controller-initiated telemetry log (refer to section 5.15.1.8) ~~which contains information about an NVM subsystem fault~~ has been generated.

...

5.15.2.2 Identify Controller data structure (CNS 01h)

<modify figure 251>

...

Figure 251: Identify – Identify Controller Data Structure

...										
539:536	O	SGL Support (SGLS): This field indicates if SGLs are supported for the NVM Command Set and the particular SGL types supported. Refer to section 4.4.								
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>...</td><td></td></tr><tr><td>17</td><td>This field specifies metadata buffer alignment requirements when CDW0.PSDT is set to 01b (refer to Figure 105). If set to '1', then use of a byte aligned contiguous physical buffer of metadata (the Metadata Pointer field in Figure 106) is supported. If cleared to '0', then use of a byte aligned contiguous physical buffer of metadata is not supported.</td></tr><tr><td>...</td><td></td></tr></table>	Bits	Description	...		17	This field specifies metadata buffer alignment requirements when CDW0.PSDT is set to 01b (refer to Figure 105). If set to '1', then use of a byte aligned contiguous physical buffer of metadata (the Metadata Pointer field in Figure 106) is supported. If cleared to '0', then use of a byte aligned contiguous physical buffer of metadata is not supported.	...	
		Bits	Description							
		...								
		17	This field specifies metadata buffer alignment requirements when CDW0.PSDT is set to 01b (refer to Figure 105). If set to '1', then use of a byte aligned contiguous physical buffer of metadata (the Metadata Pointer field in Figure 106) is supported. If cleared to '0', then use of a byte aligned contiguous physical buffer of metadata is not supported.							
...										

5.21.1.14 Timestamp (Feature Identifier 0Eh), (Optional)

<clarification on Timestamp due to NVMe 1.3 ECN 006 change>

...

The Timestamp feature enables the host to set a timestamp value in the controller. A controller indicates support for the Timestamp feature through the Optional NVM Command Support (ONCS) field in the Identify Controller data structure. The Timestamp value (refer to Figure 304Figure 303) in a Set Features command sets a timestamp value in the controller. After the current value for this feature is set, the controller updates that value as time passes. A Get Features command that requests the current value reports the timestamp value in the controller at the time the Get Features command is processed (e.g., the value set with a Set Features command for the current value plus the elapsed time since being set).

Note: If the Timestamp feature is saveable (refer to Figure 188Figure 187) and the host saves a value, then the timestamp value restored after a subsequent power on or reset event is the value that was saved (refer to section 7.8). As a result, the timestamp may appear to move backwards in time.

The accuracy of Timestamp values after initialization may be affected by vendor specific factors, such as whether the controller continuously counts after the timestamp is initialized, or whether the controller stops counting during certain intervals (e.g., non-operational power states). If the controller stops counting during such intervals, then the Synch bit in the Timestamp – Data Structure for Get Features (Figure 304) shall be set to '1'.

If the controller maintains (i.e., continues to update) the timestamp value across any type of Controller Level Reset (e.g., across a Controller Reset), then the controller shall also preserve the Timestamp Origin field (Figure 304) across that type of Controller Level Reset.

[If the controller does not maintain the timestamp across the most recent Controller Level Reset, then the Timestamp field is cleared to 0h due to that Controller Level Reset.](#)

Timestamp values should not be used for security applications. The use of the Timestamp is outside the scope of this specification.

If a Set Features command is issued for this Feature, the data structure specified in Figure 304Figure 303 is transferred in the data buffer for that command, specifying the Timestamp value.

...

5.24 Sanitize command – NVM Command Set Specific

...

5.24.1 Command Completion

<add some new status codes that Sanitize can return>

...

Figure 336: Sanitize – Command Specific Status Values

Value	Description
0Bh	Firmware Activation Requires Conventional Reset : The sanitize operation could not be started because a firmware activation is pending and a Conventional Reset is required.
10h	Firmware Activation Requires NVM Subsystem Reset : The sanitize operation could not be started because a firmware activation is pending and an NVM Subsystem Reset is required .
11h	Firmware Activation Requires Controller Level Reset : The sanitize operation could not be started because a firmware activation is pending and a Controller Level Reset is required.
20h	Namespace is Write Protected : The command is prohibited while the namespace is write protected (refer to section 8.19).
23h	Sanitize Prohibited While Persistent Memory Region is Enabled : A sanitize operation is prohibited while the Persistent Memory Region is enabled.

6.8 Flush command

<Modify a portion of section 6.8 as follows: (and see related change to 8.15.1)>

...

If a volatile write cache is not present or not enabled, then Flush commands:

- shall complete successfully and have no effect [if a sanitize operation is not in progress; and](#)
- [may complete successfully and have no effect if a sanitize operation is in progress](#).

6.14 Verify command

<modify figure 398: bits/bytes, and remove a space after “63: “>

...

Figure 398: Verify – Command Dword 10 and Command Dword 11

Bytes Bits	Description
63:00	Starting LBA (SLBA) : This field indicates the 64-bit address of the first logical block of data to be verified as part of the operation. Command Dword 10 contains bits 31:00; Command Dword 11 contains bits 63:32.

7.6.1 Initialization

<consistently: add 'Identify' >

...

7. The host should determine the configuration of the controller by issuing the Identify command, specifying the **Identify** Controller data structure. The host should then determine the configuration of each namespace by issuing the Identify command for each namespace, specifying the **Identify** Namespace data structure;

...

7.6.2 Shutdown

< deal with power loss notification status code usage>

It is recommended that the host wait a minimum of the RTD3 Entry Latency reported in the Identify Controller data structure for the shutdown operations to complete; if the value reported in RTD3 Entry Latency is 0h, then the host should wait for a minimum of one second. It is not recommended to disable the controller via the CC.EN field. This causes a Controller Reset which may impact the time required to complete shutdown processing. ~~If the host submits a new command to the controller while~~ While shutdown processing is in progress, the controller may abort any command with a status code of Commands Aborted due to Power Loss Notification.

It is safe to power off the controller when CSTS.SHST indicates shutdown processing is complete (regardless of the value of CC.EN). It remains safe to power off the controller until CC.EN transitions from '0' to '1'.

7.12.2 Traffic Based Keep Alive

<change the name of the Identify data structure>

Traffic Based Keep Alive (TBKAS) allows the host and controller to restart the Traffic Based Keep Alive Timer in the presence of Admin or I/O command processing. The controller support for TBKAS is indicated in the Controller Attributes in the Identify ~~Namespace–Controller~~ data structure (refer to Figure 251). If the Controller does not support Traffic Based Keep Alive (TBKAS is cleared to '0'), then the operation of the Keep Alive feature is described in section 7.12.1.

...

8.3.1.5 Control of Protection Information Checking – PRCHK

<correct some changes in NVMe 1.4 ECN 003 (already incorporated in NVMe 1.4b) that were not functionally correct. ECN 003 did not account for previous changes in ECN 001.>

Checking of protection information consists of the following operations performed by the controller.

- If bit 2 of the Protection Information Check (PRCHK) field of the command is set to '1', then the controller compares the protection information Guard field to the CRC-16 computed over the logical block data.
- If bit 1 of the PRCHK field is set to '1', then the controller compares unmasked bits in the protection information Application Tag field to the Logical Block Application Tag (LBAT) field in the command. A bit in the protection information Application Tag field is masked if the corresponding bit is cleared to '0' in the Logical Block Application Tag Mask (LBATM) field or the Expected Logical Block Application Tag Mask (ELBATM) field of the command.
- If bit 0 of the PRCHK field is set to '1' and the namespace is formatted for Type 1 or Type 2 protection, then the controller compares the protection Information Reference Tag field to the computed reference tag. The computed reference tag depends on the Protection Type:
 - If the namespace is formatted for Type 1 protection, the value of the computed reference tag for the first logical block of the command is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command, and the computed reference tag is incremented for each subsequent logical block. The controller shall complete the command with a status of Invalid Protection Information if the ILBRT field or the EILBRT field does not match the least significant four bytes of the SLBA field.

Note: Unlike SCSI Protection Information Type 1 protection which implicitly uses the least significant four bytes of the LBA, the controller always uses the ILBRT or EILBRT field and requires host software to initialize the ILBRT or EILBRT field to the least significant four bytes of the LBA when Type 1 protection is used.

 - If the namespace is formatted for Type 2 protection, the value of the computed reference tag for the first logical block of the command is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command, and the computed reference tag is incremented for each subsequent logical block.

<change the indentation for the following bullet to the left>

- If bit 0 of the PRCHK field is set to '1' and the namespace is formatted for Type 3 protection, ~~the value of the computed reference tag for the first LBA of the command and all subsequent logical blocks is the value contained in the Initial Logical Block Reference Tag (ILBRT) or Expected Initial Logical Block Reference Tag (EILBRT) field in the command,~~ then the controller:
 - should not compare the protection Information Reference Tag field to the computed reference tag; and
 - may ignore the ILBRT and EILBRT fields. If a command is aborted as a result of bit 0 of the PRCHK field being set to '1', then that command should be aborted with a status code of Invalid Protection Information, but may be aborted with a status code of Invalid Field in Command.

Protection checking may be disabled as a side effect of the value of the protection information Application Tag and Reference Tag fields regardless of the state of the PRCHK field in the command. If the namespace is ...

...

8.5 (Virtualization Enhancements (Optional))

...

<clarify 'pairs' >

To support the Virtualization Enhancements capability, the NVM subsystem shall support the following:

- One or more primary controllers, each of which supports:
 - One or more secondary controllers;
 - A pool of unassigned Flexible Resources that supports allocation to a primary controller and dynamic assignment to its associated secondary controllers;
 - Two or more Private Resource queue pairs;
 - Indicate support for the Virtualization Management command in the Optional Admin Command Support (OACS) field in the Identify Controller data structure;
 - The Virtualization Management command;
 - ...

8.5.1 VQ Resource Definition

<clarify 'pairs' >

...

A primary controller that supports VQ Resources shall have at least two queue pairs that are Private Resources to ensure there is a minimum of an Admin Queue pair and one I/O Queue pair for the primary controller at all times.”

8.11.1 Short Device Self-Test Operation

<modify figure 479 to change 'valid' to 'active'>

...

Figure 479: Format NVM command Aborting a Device Self-Test Operation

FNA bit ¹	NSID in Format NVM command	NSID in Device Self-test command	Abort Device Self-Test operation?
0	Any allocated NSID value (refer to section 6.1.3)	Any valid active NSID value (refer to section 6.1.4)	Yes, if the NSID values are the same
0	FFFFFFFFh	Any valid active NSID value (refer to section 6.1.4)	Yes
0	Any allocated NSID value (refer to section 6.1.3)	FFFFFFFFh	Optional
0	FFFFFFFFh	FFFFFFFFh	Yes
1	Ignored	Ignored	Yes
Key: Optional = The device self-test operation is not required to be aborted but may be aborted.			
NOTES: 1. For a Format NVM command with Secure Erase, this column refers to Bit 1 in the FNA field in the Identify Controller data structure (refer to Figure 251) and bit 0 in the FNA field is ignored. For a Format NVM command without Secure Erase, this column refers to bit 0 in the FNA field, and bit 1 in the FNA field is ignored.			

8.15 Sanitize Operations (Optional)

<add one new bullet>

...

If the Sanitize command is supported, then the NVM subsystem and all controllers shall:

- Support the Sanitize Status log page;
- Support the Sanitize Operation Completed asynchronous event;
- Support the Sanitize Operation Completed With Unexpected Deallocation asynchronous event, if the Sanitize Config feature is supported;
- Support the Exit Failure Mode action for a Sanitize command;
- Support at least one of the following sanitize operation types: Block Erase, Overwrite, or Crypto Erase; ~~and~~
- Support the same set of sanitize operation types; and
- Indicate ~~the support for all~~ supported sanitize operation types in the Sanitize Capabilities field in the Identify Controller data structure.

...

8.15.1 Sanitize Command Restrictions

<Modify a portion of section 8.15.1 as follows: (and see a related change in 6.8) >

While performing a sanitize operation and while a failed sanitize operation has occurred but successful recovery from that failure has not occurred, all enabled controllers and namespaces in the NVM subsystem are restricted to performing only a limited set of actions.

While a sanitize operation is in progress:

- All controllers in the NVM subsystem shall only process the Admin commands listed in Figure 486 subject to the additional restrictions stated in that figure;
- All I/O Commands other than a Flush command shall be aborted with a status of Sanitize In Progress;
- Processing of a Flush command is specified in section 6.8;
- Any command or command option that is not explicitly permitted in Figure 486 shall be aborted with a status of Sanitize In Progress if fetched by any controller in the NVM subsystem; and
- The Persistent Memory Region shall be prevented from being enabled (i.e., setting PMRCTL.EN to '1' does not result in PMRSTS.NRDY being cleared to '0').

...

Figure 486: Sanitize Operations – Admin Commands Allowed

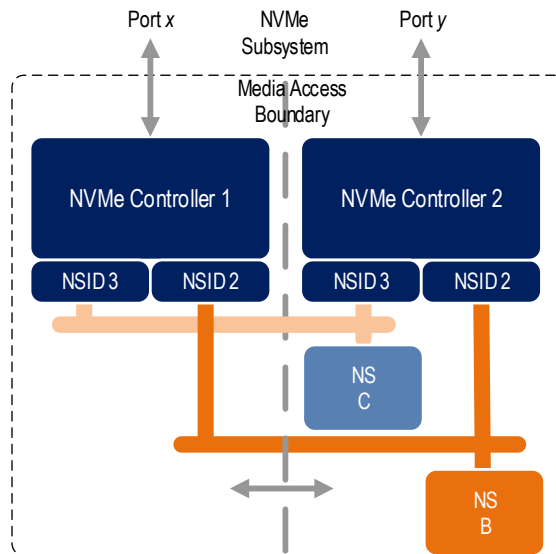
Admin Command	Additional Restrictions																
...																	
Opcode 7Fh	<div>The Fabric Commands allowed are listed below. Refer to the NVMe over Fabrics specification.<table><tr><th>Fabrics Commands</th><th>Additional Restrictions</th></tr><tr><td>Property Set</td><td></td></tr><tr><td>Connect</td><td></td></tr><tr><td>Disconnect</td><td></td></tr><tr><td>Property Get</td><td></td></tr><tr><td>Authentication Send</td><td></td></tr><tr><td>Authentication Receive</td><td></td></tr><tr><td>Vendor Specific</td><td>Commands are allowed that do not affect or retrieve user data.</td></tr></table></div>	Fabrics Commands	Additional Restrictions	Property Set		Connect		Disconnect		Property Get		Authentication Send		Authentication Receive		Vendor Specific	Commands are allowed that do not affect or retrieve user data.
Fabrics Commands	Additional Restrictions																
Property Set																	
Connect																	
Disconnect																	
Property Get																	
Authentication Send																	
Authentication Receive																	
Vendor Specific	Commands are allowed that do not affect or retrieve user data.																
Vendor Specific	Commands are allowed that do not affect or retrieve user data.																

8.20.1 Asymmetric Namespace Access Reporting Overview

...

<change 'NVMe Subsystem' to 'NVM Subsystem' in figure 495.>

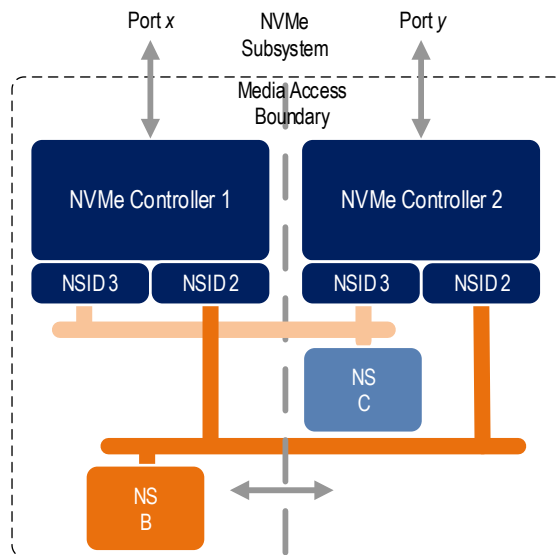
Figure 495: Namespace B and C optimized through Controller 2



...

<change 'NVMe Subsystem' to 'NVM Subsystem' in figure 496.>

Figure 496: Namespace B optimized through Controller 1



...

<change 'NVMe Subsystem' to 'NVM Subsystem' in figure 497.>

Figure 497: Multiple Namespace groups

