LEGAL NOTICE:

This NVM Express revision 1.4 technical proposal is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

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

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **"© 2007 - 2020 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

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

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

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

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**TP 4009**
Change Date	**02/24/2020**
Builds on Specification	**NVM Express 1.4**
References Other TPs	**TP4052**

Technical Proposal Author(s)

Name	Company
Fred Knight, John Meneghini	NetApp
David Black	Dell EMC
Curtis Ballard	HPE

This technical proposal defines capabilities that enable large scale environments with different domains and the possibilities of partitioning of those domains.

Capacity management is complex in a large (warehouse sized) NVM subsystem. Capacity is partitioned between various parts of the subsystem, and 100% of the capacity may not be available across the whole NVM subsystem (e.g., there may be pools of capacity in different parts of the NVM subsystem). These subsystems may also exhibit individual component failures, partial subsystem failures, or management reconfiguration actions that have NVM subsystem impact. These types of NVM subsystems are typically not guaranteed to be 100% functional vs. 100% broken.

For example, component failures may include port failures; partial failures of a controller, the failure of controller to storage capacity communication. These failures may result in an unplanned division of the NVM subsystem. Management actions include planned actions (e.g., adding or reducing NVM capacity, adding or removing ports, changing redundancy levels, reconfiguring the NVM subsystem, rolling upgrades – where controllers are upgraded at different times).

In most situations, these types of events result in ANA status codes and/or AENs.

In these complex NVM subsystems, there are additional limits beyond the total NVM subsystem capacity. For example, a system with multiple RAID sets where any individual namespace is limited to the capacity of the RAID set on which that namespace is placed, rather than the total NVM subsystem capacity.

This TP addresses the requirements for these types of systems.

Revision History

Revision Date	Change Description
5/31/17	Break into 2 parts – ANA and Partition Model (this part)
2/6/18	Restart
12/18/18	Move model text into section 7. Add Firmware Update changes. Convert all notes to text proposals. Open questions remain in <green> text.
2/12/19	Update Log page scope descriptions (for domain vs. subsystem), add more sanitize description, add “empty” domain, clarify impact on virtualization (primary/secondary must be in same domain). All green text actions completed.
2/14/19	Add “Subsystem is Divided” error for Sanitize.
3/7/19	Add requirement that multi-domain support requires ANA support.
3/19/19	Re-baseline against rev 1.4
3/26/19	Adjust language about returned lists (IDENTIFY) so that rather than returning just information about the domain containing the controller processing the command, that information is returned about objects that are accessible by the controller processing the command. This solves the problem of returning information about domains that do not contain controllers.
4/7/19	Add TEGCAP and UEGCAP acronyms. 1st pass at “accessible” definition in language in Identify command (CNS types). Added NS E (in the domains picture) to illustrate the need for MEGCAP (also added the MEGCAP field).
4/25/19	Merge with latest (4/25) TP4052 and rewrite Capacity Management section to point to all the different layers of capacity values. Update Namespace Management section to point to new Capacity Management section.
4/29/19	Add definition of 0h to Endurance Group Log capacity values (for backward compatibility). Subsystem is Divided (new) replaced with Internal Path Error (existing). Fix introduction paragraph for Identify Command Overview. Expand descriptions in Figure CAP_USE_1 to better explain how to use this table. Update the description on Host use of Domain Identifiers.
5/8/19	Replace “Divided” error with ANA errors.
5/21/19	Move Capacity Management to TP4052; make sure Endurance Groups are optional.
6/3/19	Update description of NSID=FFFFFFFFh case in Figure 494 when some ANA Groups are “not available”, but some ANA Groups are “available”. The feature may get set on some, and not set on others – depending on the ANA state.
6/18/19	Address 30-day review concerns - update NVM Subsystem Reset text to better account for domains (includes updating 7.3.1 references to point to 7.3 – to refer to both subsystem and domain reset). Make PMR disable text the same for both forms of NVMe Subsystem Reset (fixes cut/paste error).
6/27/19	Add allowance for domain reset to also reset the whole NVM subsystem. Remove domain scope from the Persistent Error log.
7/5/19	Update merge with rev 1.4.
8/15/19	Apply updates to Figure 242 so it matches the normative text in section 5.15.2.99.
2/11/2020	Ratification ready
2/12/2020	Editorial comments by HPE – Re-Ratification ready
2/24/2020	Ratified

Black text indicates existing material in the NVMe Base 1.4 specification, Red text indicates material added by this TP, and Blue text indicates material that is found in TP4052.

Description of Specification Changes

Add the following sections as shown below:

1.4 Theory of Operation

…

An NVM Express controller is associated with a single PCI Function. The capabilities and settings that apply to the entire controller are indicated in the Controller Capabilities (CAP) register and the Identify Controller data structure.

An NVM Subsystem is made up of a single domain or multiple domains as described in section 7.NEW.

A namespace is a quantity of non-volatile memory that may be formatted into logical blocks.

…

1.6.TBD Domain

A domain is the smallest indivisible unit that shares state (e.g., power state, capacity information).

…

1.6.15 firmware slot

A firmware slot is a location in ~~the NVM subsystem~~ a domain used to store a firmware image. The ~~NVM subsystem~~ domain stores from one to seven firmware images. Controllers in the same domain share the same firmware slots.

…

1.6.28 NVM subsystem

An NVM subsystem includes one or more domains, 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.

…

3.1.1 Offset 0h: CAP – Controller Capabilities

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

Figure 69: Offset 0h: CAP – Controller Capabilities

Bits	Type	Reset	Description
…			
36	RO	Impl Spec	**NVM Subsystem Reset Supported (NSSRS):** This bit indicates whether the controller supports the NVM Subsystem Reset feature defined in section 7.3.~~1~~. This bit is set to '1' if the controller supports the NVM Subsystem Reset feature. This bit is cleared to '0' if the controller does not support the NVM Subsystem Reset feature.
…			

…

3.1.6 Offset 1Ch: CSTS – Controller Status

Figure 79: Offset 1Ch: CSTS – Controller Status

Bits	Type	Reset	Description
…			
04	RWC	HwInit	**NVM Subsystem Reset Occurred (NSSRO):** The initial value of this bit is set to '1' if the last occurrence of an NVM Subsystem Reset (refer to section 7.3) occurred while power was applied to the ~~NVM subsystem~~ domain. The initial value of this bit is cleared to '0' following an NVM Subsystem Reset due to application of power to the ~~NVM subsystem~~ domain. This bit is only valid if the controller supports the NVM Subsystem Reset feature defined in section 7.3~~.1~~ as indicated by CAP.NSSRS set to '1'. The reset value of this bit is cleared to '0' if an NVM Subsystem Reset causes activation of a new firmware image.
…			

3.1.7 Offset 20h: NSSR – NVM Subsystem Reset

This optional register provides host software with the capability to initiate an NVM Subsystem Reset. Support for this register is indicated by the state of the NVM Subsystem Reset Supported (CAP.NSSRS) field. If the register is not supported, then the address range occupied by the register is reserved. Refer to section 7.3~~.1~~.

…

4.6.1.2.4 Path Related Status Definition

Completion queue entries with a Status Code type of Path Related Status (refer to Figure 132) indicate

…

Figure 132: Status Code – Path Related Status Values

Value	Description
00h	**Internal Path Error:** The command was not completed as the result of a controller internal error that is specific to the controller processing the command. Retries for the request function should be based on the setting of the DNR bit (refer to Figure 124).
01h	**Asymmetric Access Persistent Loss:** The requested function (e.g., command) is not able to be performed as a result of the relationship between the controller and the namespace, NVM Set, or Endurance Group being in the ANA Persistent Loss state (refer to section 8.20.3.4). The command should not be re-submitted to the same controller.
02h	**Asymmetric Access Inaccessible:** The requested function (e.g., command) is not able to be performed as a result of the relationship between the controller and the namespace, NVM Set, or Endurance Group being in the ANA Inaccessible state (refer to section 8.20.3.3). The command should not be re-submitted to the same controller.
03h	**Asymmetric Access Transition:** The requested function (e.g., command) is not able to be performed as a result of the relationship between the controller and the namespace, NVM Set, or Endurance Group transitioning between Asymmetric Namespace Access states (refer to section 8.20.3.5). The requested function should be retried after the transition is complete.
04h to 5Fh	Reserved
Controller detected Pathing errors	
60h	**Controller Pathing Error:** A pathing error was detected by the controller.
61h to 6Fh	Reserved
Host detected Pathing errors	
70h	**Host Pathing Error:** A pathing error was detected by the host.

Figure 132: Status Code – Path Related Status Values

Value	Description
71h	**Command Aborted By host:** The command was aborted as a result of host action (e.g., the host disconnected the Fabric connection).
72h to 7Fh	Reserved
Other Pathing errors	
80h to BFh	I/O Command Set Specific
C0h to FFh	Vendor Specific

…

Make changes to the sections as shown below:

5.11 Firmware Commit command

NOTE: This command was known in NVM Express revision 1.0 and 1.1 as “Firmware Activate.”

The Firmware Commit command is used to modify the firmware image or Boot Partitions.

When modifying a firmware image, the Firmware Commit command verifies that a valid firmware image has been downloaded and commits that revision to a specific firmware slot. The host may select the firmware image to activate on the next Controller Level Reset as part of this command. The host may determine the currently executing firmware revision by examining the Firmware Revision field in the Identify Controller data structure in Figure 247. The host may determine the firmware revision to be executed on the next Controller Level Reset by examining the Firmware Slot Information log page. All controllers in t~~he NVM subsystem~~ a domain share firmware slots and the same firmware image is applied to all controllers in that domain (i.e., all the controllers in the NVM subsystem if multiple domains are not supported or all the controllers in that domain if multiple domains are supported).

Activation of a firmware image may result in a change in controller behavior that is not expected by the host (e.g., an incompatible change in the UUID List (refer to section 8.24.2)). In this case, if the Commit Action field is set to 011b, then the controller shall abort the command with a status of Firmware Activation Requires Conventional Reset.

…

5.14.1 Log Specific Information

Figure 191 and Figure 192 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 Domain return information that is global to the Domain. 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, support for multiple domains or 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 191 and Figure 192), 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 105 apply.

Figure 191: Get Log Page – Log Page Identifiers

Log Identifier	Scope	Log Page Name	Reference Section
00h	Reserved		
01h	Controller	Error Information	5.14.1.1
02h	Controller [1] Namespace [2]	SMART / Health Information	5.14.1.2
03h	Domain / NVM subsystem [6]	Firmware Slot Information	5.14.1.3
04h	Controller	Changed Namespace List	5.14.1.4
05h	Controller	Commands Supported and Effects	5.14.1.5
06h	Controller [3] Domain / NVM subsystem [4, 6]	Device Self-test [5]	5.14.1.6
07h	Controller	Telemetry Host-Initiated [5]	5.14.1.7
08h	Controller	Telemetry Controller-Initiated [5]	5.14.1.8
09h	Domain / NVM subsystem [6]	Endurance Group Information	5.14.1.9
0Ah	Domain / NVM subsystem [6]	Predictable Latency Per NVM Set	5.14.1.10
0Bh	Domain / NVM subsystem [6]	Predictable Latency Event Aggregate	5.14.1.11
0Ch	Controller	Asymmetric Namespace Access	5.14.1.12
0Dh	NVM subsystem	Persistent Event Log [5]	5.14.1.13
0Eh	Controller	LBA Status Information	5.14.1.14
0Fh	Domain / NVM subsystem [6]	Endurance Group Event Aggregate	5.14.1.15
10h	Domain / NVM Subsystem [6]	Media Unit Status	5.14.1.TBD
11h	Domain / NVM subsystem [6]	Media Unit Supported Configuration List	5.14.1.TBD1
12h – 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 [5]		

KEY:
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.
3. Bit 0 is cleared to '0' in the DSTO field in the Identify Controller data structure (refer to Figure 111).
4. Bit 0 is set to '1' in the DSTO field in the Identify Controller data structure.
5. Selection of a UUID may be supported. Refer to section 8.24
6. For NVM subsystems that support multiple domains (refer to the MDS bit in the Identify Controller data structure), Domain scope information is returned.

Figure 192: Get Log Page – Log Page Identifiers, NVM Command Set Specific

Log Identifier	Scope	Description	Reference Section
80h	Controller	Reservation Notification	5.14.1.9.1
81h	NVM subsystem	Sanitize Status	5.14.1.9.2
82h – BFh	Reserved		

KEY:
O/M definition: O = Optional, M = Mandatory

...

The following information is modified …:

5.14.1.9 Endurance Group Information (Log Identifier 09h)

This log page is used to provide endurance information based on the Endurance Group (refer to section 8.17). An Endurance Group contains capacity that may be allocated to ~~consists of~~ zero or more NVM Sets. Capacity that has not been allocated to an NVM Set is unallocated Endurance Group capacity. The information provided is over the life of the Endurance Group. The Endurance Group Identifier is specified in the Log Specific Identifier field in Command Dword 11 of the Get Log Page command. The log page is 512 bytes in size.

Figure 204: Get Log Page – Endurance Group Log (Log Identifier 09h)

Bytes	Description
…	
05	**Percentage Used:** Contains a vendor specific estimate of the percentage of life used for the Endurance Group based on the actual usage and the manufacturer's prediction of NVM life. A value of 100 indicates that the estimated endurance of the NVM in the Endurance Group has been consumed but may not indicate an NVM failure. The value is allowed to exceed 100. Percentages greater than 254 shall be represented as 255. This value shall be updated once per power-on hour when the controller is not in a sleep state. Refer to the JEDEC JESD218A standard for SSD device life and endurance measurement techniques.
07:06	**Domain Identifier:** This field indicates the identifier of the domain that contains this Endurance Group. If the NVM subsystem supports multiple domains, this field shall be set to a non-zero value. If cleared to 0h, the NVM subsystem does not support multiple domains.
31:08	Reserved
…	
159:144	**Number of Error Information Log Entries:** Contains the number of Error Information log entries over the life of the controller for the Endurance Group.
175:160	**Total Endurance Group Capacity (TEGCAP):** This field indicates the total NVM capacity in this Endurance Group. The value is in bytes. If this field is cleared to 0h, the NVM subsystem does not report the total NVM capacity in this Endurance Group.
191:176	**Unallocated Endurance Group Capacity (UEGCAP):** This field indicates the unallocated NVM capacity in this Endurance Group. The value is in bytes. If this field is cleared to 0h, the NVM subsystem does not report the unallocated NVM capacity in this Endurance Group.
511:160	Reserved

…

The following text is modified based on material in TP4052 (Endurance Group Management):

5.14.1.TBD Media Unit (Log Identifier 10h)

This log page is used to describe the configuration and wear of Media Units (refer to section 8.TBD). …

<Refer to TP4052>

…

5.14.2.TBD Media Unit Supported Configuration List (Log Identifier 11h)

This log page is used provide a list of Media Unit Supported Configuration Descriptors …

<Refer to TP4052>

…

The following information is modified …

5.15 Identify command

5.15.1 Identify command overview

The Identify command returns a data buffer that describes information about the NVM subsystem, the domain, the controller or the namespace(s). The data structure is 4096 bytes in size.

…

Figure 242: Identify – Command Dword 11

Bits	Description
31:16	Reserved
15:00	**NVM Set Identifier (NVMSETID):** This field specifies the identifier of the NVM Set or the Domain. This field is used for Identify operations with a CNS value of 04h and 18h. This field should be cleared to 0h for Identify operations with other CNS values.

…

Figure 244: Identify – CNS Values

CNS Value	O/M 1	Definition	NSID[2]	CNTID[3]	Reference Section
Active Namespace Management					
…					
Controller and Namespace Management					
…					
16h	O	A Namespace Granularity List (refer to Figure 255) is returned to the host for up to sixteen Namespace Granularity Entries.	N	N	5.15.2.12
17h	O	A UUID List (refer to Figure 257) is returned to the host.	N	N	5.15.2.13
19h	O [7]	Endurance Group List.	N	N	5.15.2.TBD1
18h	O [8]	Domain List	N	N	5.15.2.99
18h to 1Fh		Reserved			

Figure 244: Identify – CNS Values

CNS Value	O/M [1]	Definition	NSID [2]	CNTID [3]	Reference Section
Future Definition					
20h to FFh		Reserved			

NOTES:
1. O/M definition: O = Optional, M = Mandatory.
2. The NSID field is used: Y = Yes, N = No.
3. The CDW10.CNTID field is used: Y = Yes, N = No.
4. Mandatory for controllers that support the Namespace Management capability (refer to section 8.12).
5. Mandatory for controllers that support Virtualization Enhancements (refer to section 8.5).
6. Selection of a UUID may be supported. Refer to section 8.24.
7. Mandatory for controllers that support Capacity Endurance Group Management (refer to section 8.TBD in TP4052).
8. Mandatory for controllers that support Endurance Group Management (refer to section 8.TBD in TP4052) in an NVM subsystem that supports multiple domains (refer to section 7.NEW).

5.15.2.2 Identify Controller data Structure (CNS 01h)

The Identify Controller data structure (refer to Figure 111) is returned to the host for this controller.

Figure 247: Identify – Identify Controller Data Structure

Bytes	O/M	Description
Controller Capabilities and Features		
…		
71:64	M	**Firmware Revision (FR):** Contains the currently active firmware revision for the ~~NVM subsystem~~ domain of which this controller is a part. This is the same revision information that may be retrieved with the Get Log Page command, refer to section 5.14.1.3. Refer to section 1.5 for ASCII string requirements.
…		
75:73	M	**IEEE OUI Identifier (IEEE):** Contains the Organization Unique Identifier (OUI) for the controller vendor. The OUI shall be a valid IEEE/RAC assigned identifier that may be registered at http://standards.ieee.org/develop/regauth/oui/public.html.
…		
99:96	M	**Controller Attributes (CTRATT):** This field indicates attributes of the controller. Bits 31:11~~10~~ are reserved. Bit 10 **(Multi-Domain Subsystem) (MDS):** If set to '1', then the NVM subsystem supports the multiple domains (refer to section 7.NEW). If cleared to '0', then the NVM subsystem does not support the reporting of multiple domains and the NVM subsystem consists of a single domain. Bit 9 **(UUID List):** If set to '1', then the controller supports reporting of a UUID List (refer to Figure 257). If cleared to '0', then the controller does not support reporting of a UUID List (refer to section 8.24). … Bit 0 if set to '1' then the controller supports a 128-bit Host Identifier. Bit 0 if cleared to '0' then the controller does not support a 128-bit Host Identifier.
…		

Bytes	O/M	Description
260	M	**Firmware Updates (FRMW):** This field indicates capabilities regarding firmware updates. Refer to section 8.1 for more information on the firmware update process. Bits 7:5 are reserved. Bit 4 if set to '1' indicates that the controller supports firmware activation without a reset. If cleared to '0', then the controller requires a reset for firmware to be activated. Bits 3:1 indicate the number of firmware slots ~~that~~ supported by the domain that contains this controller ~~supports~~. This field shall specify a value between one and seven, indicating that at least one firmware slot is supported and up to seven maximum. This corresponds to firmware slots 1 through 7. Bit 0 if set to '1' indicates that the first firmware slot (slot 1) is read only. If cleared to '0', then the first firmware slot (slot 1) is read/write. Implementations may choose to have a baseline read only firmware image.
…		
295:280	O	**Total NVM Capacity (TNVMCAP):** This field indicates the total NVM capacity ~~in the NVM subsystem~~ that is accessible by the controller. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to section 8.12) is supported. Refer to section 7.TBD (Capacity Management in TP4052).
311:296	O	**Unallocated NVM Capacity (UNVMCAP):** This field indicates the unallocated NVM capacity ~~in the NVM subsystem~~ that is accessible by the controller. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to section 8.12) is supported. Refer to section 7.TBD (Capacity Management in TP4052).
…		
357:356	O	**Domain Identifier:** This field indicates the identifier of the domain (refer to section 7.NEW) that contains this controller. If the MDS bit is set to '1', then this field shall be set to a non-zero value. If the NVM subsystem does not support multiple domains (i.e., the NVM subsystem consists of a single domain), then this field shall be cleared to 0h.
…		
383:368	O	**Max Endurance Group Capacity (MEGCAP):** This field indicates the maximum capacity of a single Endurance Group. If this field is cleared to 0h, the NVM subsystem does not report a maximum Endurance Group Capacity value.
…		
NVM Command Set Attributes		
…		

Bytes	O/M	Description
524	M	**Format NVM Attributes (FNA):** This field indicates attributes for the Format NVM command. Bits 7:3 are reserved. Bit 2 indicates whether cryptographic erase is supported as part of the secure erase functionality. If set to '1', then cryptographic erase is supported. If cleared to '0', then cryptographic erase is not supported. Bit 1 indicates whether the secure erase functionality applies to all namespaces in ~~an~~ the NVM subsystem or is specific to a particular namespace. If set to '1', then any secure erase performed as part of a format operation results in a secure erase of all namespaces in the NVM subsystem. If cleared to '0', then any secure erase performed as part of a format results in a secure erase of the particular namespace specified. Bit 0 indicates whether the format operation (excluding secure erase) applies to all namespaces in ~~an~~ the NVM subsystem or is specific to a particular namespace. If set to '1', then all namespaces in ~~an~~ the NVM subsystem shall be configured with the same attributes and a format (excluding secure erase) of any namespace results in a format of all namespaces in ~~an~~ the NVM subsystem. If cleared to '0', then the controller supports format on a per namespace basis.
…		
531	M	Reserved
535:TBD	M	Reserved

…

5.15.2.5 NVM Set List (CNS 04h)

Figure 250 defines an NVM Set List. The data structure is an ordered list by NVM Set Identifier, starting with the first NVM Set Identifier supported by the NVM subsystem that is equal to or greater than the NVM Set Identifier indicated in CDW11.NVMSETID and are accessible by the controller processing the command. The NVM Set List describes the attributes for each NVM Set in the list based on the NVM Set Attributes Entry in Figure 250.

…

5.15.2.8 Namespace Attached Controller List (CNS 12h)

A Controller List (refer to section 4.11) of up to 2,047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers of controllers that are attached to the namespace specified in the Namespace Identifier (NSID) field. If the NSID field is set to FFFFFFFFh, then the controller should fail the command with a status code of Invalid Field in Command.

5.15.2.9 Controller List (CNS 13h)

A Controller List (refer to section 4.11) of up to 2,047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers of controllers in the NVM subsystem that ~~may or may not be~~ are capable of being attached to namespace(s).

…

5.15.2.X Endurance Group List (19h)

An Endurance Group List (refer to Figure EGIL) of up to 2,047 Endurance Group identifiers …

<Note: Refer to TP4052>

…

5.15.2.TBD Domain List (CNS 18h)

Figure defines a Domain List. The data structure is an ordered list by Domain Identifier, starting with the first Domain Identifier that is equal to or greater than the Domain Identifier specified in CDW11.NVMSETID and is accessible by the controller processing the command. The Domain List describes the attributes for each Domain in the list based on the Domain Attributes Entry in Figure .

Figure 9998 Domain List

Bytes	Description
00	**Number of Identifiers:** This field indicates the number of Domain Attributes Entries in the list. There are up to 31 entries in the list. A value of 0h indicates that there are no entries in the list.
127:01	Reserved
255:128	**Entry 0:** This field contains the first Domain Attributes Entry in the list, if present.
383:256	**Entry 1:** This field contains the second Domain Attributes Entry in the list, if present.
…	**…**
(N*128+255): (N*128+128)	**Entry N:** This field contains the N+1 Domain Attributes Entry in the list, if present.

Figure 9999 Domain Attributes Entry

Bytes	Description
01:00	**Domain Identifier:** This field indicates the identifier of the Domain accessible by the controller that is described by this entry.
15:02	Reserved
31:16	**Total Domain Capacity:** This field indicates the total NVM capacity in this Domain. The value is in bytes.
47:32	**Unallocated Domain Capacity:** This field indicates the unallocated NVM capacity in this Domain. The value is in bytes.
63:48	**Max Endurance Group Domain Capacity:** This field indicates the maximum capacity of a single Endurance Group in this Domain. If this field is cleared to 0h, the NVM subsystem does not report a maximum Endurance Group Domain Capacity value.
127:64	Reserved

…

5.23 Format NVM command – NVM Command Set Specific

…

The scope of the format operation and the scope of the format with secure erase depend on the attributes that the controller supports for the Format NVM command and the Namespace Identifier specified in the command as described in Figure 327. The type of secure erase, if applicable, is based on the setting of the Secure Erase Settings field in Command Dword 10 as defined in Figure 328.

If the NVM subsystem supports multiple domains and the Format NVM command is not able to format the specified namespaces as a result of the NVM subsystem being divided (refer to section 7.NEW), then the Format NVM command shall be aborted with a status code of Asymmetric Access Inaccessible or Asymmetric Access Persistent Loss.

…

5.24 Sanitize command – NVM Command Set Specific

The Sanitize command is used to start a sanitize operation or to recover from a previously failed sanitize operation. The sanitize operation types that may be supported are Block Erase, Crypto Erase, and Overwrite. All sanitize operations are processed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation). Refer to section 8.15 for details on the sanitize operation.

If the NVM subsystem supports multiple domains and the Sanitize command is not able to start a sanitize operation as a result of the NVM subsystem being divided (refer to section 7.NEW), then the Sanitize command shall be aborted with a status code of Asymmetric Access Inaccessible or Asymmetric Access Persistent Loss.

When a sanitize operation starts on any controller, all controllers in the NVM subsystem:

…

7.3 Resets

7.3.1 NVM Subsystem Reset in Single Domain NVM Subsystems

The scope of an NVM Subsystem Reset depends on whether the NVM Subsystem supports multiple domains. In an NVM Subsystem that does not support multiple domains, the scope of the NVM Subsystem Reset is the entire NVM Subsystem.

An NVM Subsystem Reset is initiated when:

...

7.3.2 NVM Subsystem Reset in Multiple Domain NVM Subsystems

The scope of an NVM Subsystem Reset depends on whether the NVM Subsystem supports multiple domains. In an NVM Subsystem that supports multiple domains, the scope of the NVM Subsystem Reset is either the controllers that are in a domain or the entire NVM Subsystem.

An NVM Subsystem Reset on a domain is initiated when:

- Power is applied to that domain;
- A value of 4E564D65h (i.e., “NVMe”) is written to the NSSR.NSSRC field of one of the controllers in that domain; or
- A vendor specific event occurs within that domain.

When an NVM Subsystem Reset occurs the entire domain is reset. This includes the initiation of a Controller Level Reset on all controllers that are in the domain, disabling of the Persistent Memory Region associated with all controllers that are in the domain, and a transition to the Detect LTSSM state by all PCI Express ports that are in the domain.

Alternatively, an NVM Subsystem Reset in an NVM Subsystem that supports multiple domains may reset the entire NVM Subsystem.

The occurrence of an NVM Subsystem Reset while power is applied to the domain is reported by the initial value of the CSTS.NSSRO field following the NVM Subsystem Reset. This field may be used by host software to determine if the sudden loss of communication with a controller was due to an NVM Subsystem Reset or some other condition.

The ability for host software to initiate an NVM Subsystem Reset by writing to the NSSR.NSSRC field is an optional capability of a controller indicated by the state of the CAP.NSSRS field. An implementation may protect the domain from an inadvertent NVM Subsystem Reset by not providing this capability to one or more controllers that are in the domain.

…

7.3.3 Controller Level Reset

…

7.TBD Capacity Management

NVM Subsystems may report capacity-related information …

<Note: Refer to TP4052>

7.8 Feature Values

The Get Features command, (refer to section 5.13), and Set Features command, (refer to section 5.21), may be used to read and modify operating parameters of the controller.

…

For feature values that apply to a namespace:

a) if the NSID field is set to an active namespace identifier (refer to section 6.1), then:

- the Set Features command shall set the specified feature value of the specified namespace; and
- the Get Features command shall return the current setting of the requested feature value for the specified namespace;

b) if the NSID field is set to FFFFFFFFh, then:

- for the Set Features command, the controller shall:
  - if the MDS bit is set to '1' in the Identify Controller data structure, abort the command with Invalid Field in Command; or
  - ~~the Set Features command shall~~ if the MDS bit is cleared to '0' in the Identify Controller data structure, unless otherwise specified, set the specified feature value for all namespaces attached to the controller processing the command;

and

- for the Get Features command, the controller shall, unless otherwise specified in section 5.21.1, ~~fail~~ abort the command with a status code of Invalid Namespace or Format;

and

c) if the NSID field is set to any other value, then the Set Features command and the Get Features command shall fail as described in the rules for namespace identifier usage in Figure 105.

…

7.NEW Domains and Divisions

7.NEW.1 Overview

An NVM subsystem may be made up of a single domain or multiple domains (i.e., two or more). A domain is the smallest indivisible unit that shares state (e.g., power state, capacity information). An NVM subsystem that supports multiple domains shall support Asymmetric Namespace Access Reporting (refer to section 8.20).

A common example of a simple implementation of an NVM subsystem is one that consists of a single domain (i.e., multiple domains are not supported).

Each domain is independent, and the boundaries between domains are communication boundaries (e.g., fault boundaries, management boundaries). If multiple domains are present in an NVM subsystem, then those domains cooperate in the operation of that NVM subsystem. If a domain is unable to cooperate in the operation of the NVM subsystem, then the NVM subsystem has become divided.

A division is an event (e.g., failure of a domain) or action (e.g. management action or reconfiguration) within the NVM subsystem that affects communication between the domains contained in the NVM subsystem (refer to Figure 9996 and Figure 9997). If a division exists, global state within the NVM subsystem may be impacted (e.g., a controller may only have information about the state of the domains with which the controller is able to communicate). A division event or action may:

- affect access to namespaces (refer to section 8.20); or
- impact operations that have NVM subsystem scope (e.g., TNVMCAP, Sanitize, Format, SMART information).

A domain is comprised of:

- zero or more controllers; and
- zero or more NVM Endurance Groups.

If an NVM subsystem supports multiple domains, then all controllers in that NVM subsystem shall:

- set the MDS bit to ‘1’ in the CTRATT field in the Identify Controller data structure (refer to Figure 111);
- set the Domain Identifier in each Endurance Group descriptor, if supported, to a non-zero value; and
- set the Domain Identifier in each Identify Controller data structure to a non-zero value.

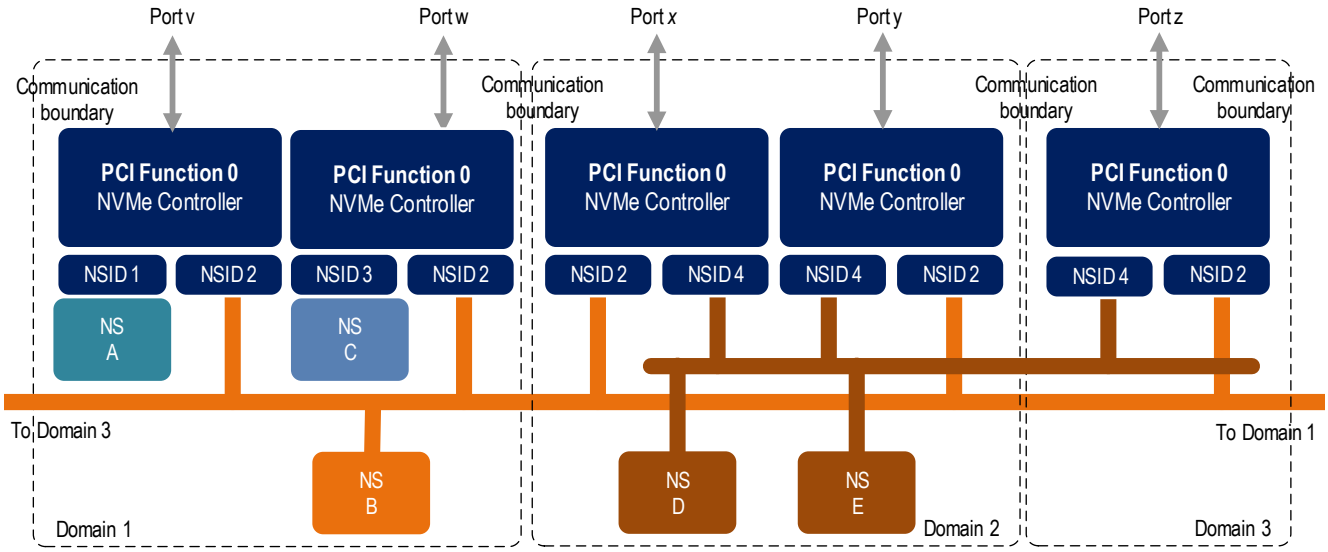
If an NVM subsystem supports multiple domains, then controllers in that NVM subsystem may:

- support Endurance Groups (refer to Endurance Groups bit in the CTRATT field of Identify Controller data structure).

For NVM subsystems that support multiple domains, each domain shall be assigned a domain identifier that is unique within the NVM subsystems (refer to the Domain Identifier field in Figure 245 and section 7.NEW.3). For NVM subsystems that do not support multiple domains, Domain Identifier fields are cleared to 0h.

Figure 9996 shows an example of an NVM subsystem that consists of three domains. Domain 1 contains two controllers and some amount of NVM storage capacity which has been allocated to two private namespaces (i.e., NS A and NS C) and a shared namespace (i.e., NS B). Domain 2 contains two controllers and some amount of NVM storage capacity which has been allocated to two shared namespaces (i.e., NS D and NS E). Domain 3 contains one controller, and no NVM storage capacity.

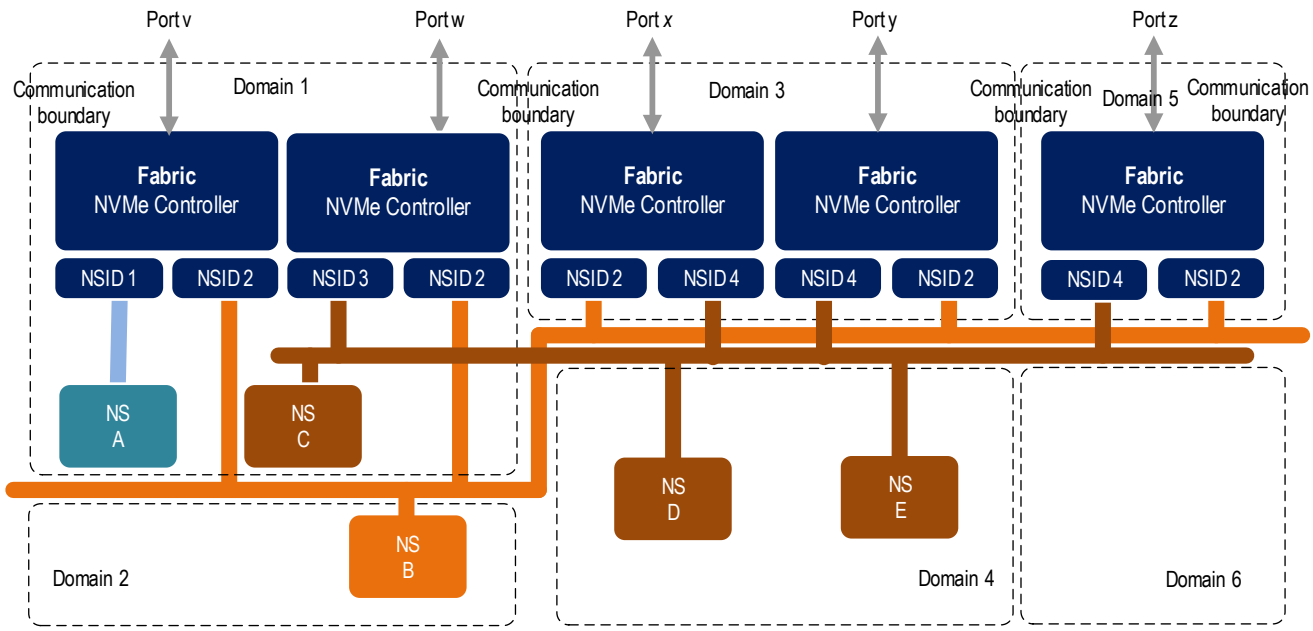
Figure 9996: Example 1 Domain Structure



If, in the example shown in Figure 9996, a division event occurs that results in Domain 1 no longer being able to communicate with Domain 2 and Domain 3, then the NVM subsystem would consist of two parts. The first part consists of Domain 1 and the second part consists of Domain 2 and Domain 3.

Figure 9997 shows an example of an NVM subsystem that consists of six domains, of which, three are domains that contain controllers. Domain 1 is a domain that contains two controllers and some amount of NVM storage capacity from which NVM Endurance Groups have been created that contain a private namespace (i.e., NS A) and a shared namespace (i.e., NS C). Domain 2 is a domain that contains no controllers and contains some amount of NVM storage capacity from which NVM Endurance Groups have been created that contain a shared namespace (i.e., NS B). Domain 3 is a domain that contains two controllers and no NVM storage capacity. Domain 4 is a domain that contains no controllers and contains some amount of NVM storage capacity from which NVM Endurance Groups have been created that contain two shared namespaces (i.e., NS D and NS E). Domain 5 is a domain that contains one controller and no NVM storage capacity. Domain 6 is a domain that contains no controllers and no NVM storage capacity allocated to an NVM Endurance Group (i.e., an empty domain).

Figure 9997: Example 2 Domain Structure



Key: - - - - - (dashed line) – communication boundary

7.NEW.2 Domains and Reservations

If an NVM subsystem supports multiple domains and Persistent Reservations (refer to section 8.8), then resumption after a division event (e.g., resumption of operation, resumption of communication) requires that all persistent reservation state within the domains in the NVM subsystem that are no longer divided be synchronized (i.e., updated).

If the reservation state for a namespace is not synchronized, then the ANA Group that contains that namespace shall transition to the ANA Inaccessible state (refer to section 8.20.3.3) and remain in that state until the Persistent Reservation state is synchronized. If the Persistent Reservation state is not able to be synchronized, then:

- ANA Persistent Loss state may be returned as described in section 8.20.3.4; or
- the controller may stop processing commands and set the Controller Fatal Status (CSTS.CFS) bit to '1' (refer to section 10.5).

7.NEW.3 Domain Identifier Use (Informative)

Domain Identifier values indicate the parts of the NVM subsystem that comprise a domain.

The host may use these values to determine which Endurance Groups (refer to section 8.17) are contained in the same domain and which are contained in a different domain. Examples of host use of the domain identifier include:

- host data redundancy software (e.g., RAID) that may use the Endurance Group's Domain Identifier to determine which Endurance Groups may fail together (e.g., Endurance Groups in the same domain) and which Endurance Groups may fail independently (e.g., Endurance Groups in different domains); and
- host application software may use the controller's Domain Identifier to determine which controllers share domains (e.g., controllers that may fail together) and which controllers are a part of different domains (e.g., controllers that may fail independently).

…

8.5 Virtualization Enhancements (Optional)

Virtualized environments may use an NVM subsystem with multiple controllers to provide virtual or physical hosts direct I/O access. The NVM subsystem is composed of primary controller(s) and secondary controller(s), where the secondary controller(s) depend on primary controller(s) for dynamically assigned resources. A host may issue the Identify command to a primary controller specifying the Secondary Controller List to discover the secondary controllers associated with that primary controller. All secondary controllers shall be part of the same domain as the primary controller with which they are associated.

Controller resources may be assigned or removed from a controller using the Virtualization Management command issued to a primary controller. The following types of controller resources are defined:

8.8 Reservations (Optional)

NVM Express reservations provide capabilities that may be utilized by two or more hosts to coordinate access to a shared namespace. The protocol and manner in which these capabilities are used is outside the scope of this specification. Incorrect application of these capabilities may corrupt data and/or otherwise impair system operation.

Reservation operation after a division event (refer to section 7.NEW.1) is described in section 7.NEW.2.

A reservation on a namespace restricts hosts access to that namespace.

…

8.12 Namespace Management (Optional)

…

The size of a namespace is based on the number of logical blocks requested in a create operation, the format of the namespace, and any characteristics (e.g., endurance). The controller determines the NVM capacity allocated for that namespace. Namespaces may be created with different usage characteristics (e.g., endurance) that utilize differing amounts of NVM capacity. Namespace characteristics and the mapping of these characteristics to NVM capacity usage are outside the scope of this specification.

~~The total and unallocated NVM capacity for the NVM subsystem is reported in the Identify Controller data structure (refer to Figure 247). For controllers that support NVM Sets, the total and unallocated NVM capacity for each NVM Set is reported as part of the NVM Set Attributes Entry (refer to Figure 251)~~. Reporting of capacity information for the NVM subsystem, Domain, Endurance Group, and NVM Set are described in section 7.TBD (Capacity Management in TP4052). For each namespace, the NVM Set in which the namespace is allocated is reported in the Identify Namespace data structure. The NVM Set to be used for a namespace is based on the value in the NVM Set Identifier field in a create operation. If the NVM Set Identifier field is cleared to 0h in a create operation, then the controller shall choose the NVM Set from which to allocate the namespace.

For each namespace, the NVM capacity used for that namespace is reported in the Identify Namespace data structure (refer to Figure 245). The controller may allocate NVM capacity in units such that the requested size for a namespace may be rounded up to the next unit boundary. The units in which NVM capacity is allocated

are reported in the Namespace Granularity List (refer to Figure 255), if supported. For example, if host software requests a namespace of 32 logical blocks with a logical block size of 4 KiB for a total size of 128 KiB and the allocation unit for the implementation is 1 MiB, then the NVM capacity consumed may be rounded up to 1 MiB. The NVM capacity fields may not correspond to the logical block size multiplied by the total number of logical blocks.

The method of allocating ANA Group identifiers is outside the scope of this specification. If the ANA Group Identifier (refer to Figure 262) is cleared to 0h, then the controller shall determine the ANAGRPID that is assigned to that namespace.

To create a namespace, host software performs the following actions:

1. Host software requests the Identify Namespace data structure that specifies common namespace capabilities (Identify with a setting of CDW1.NSID set to FFFFFFFFh and CNS cleared to 0h);
2. If the controller supports reporting of Namespace Granularity, host software optionally requests the Namespace Granularity List defined in Figure 255 (Identify command with CNS set to 16h).
3. Host software determines available capacity (refer to section 7.TBD (Capacity Management in TP4052));
4. Host software creates the data structure defined in Figure 270. Host software sets the host software specified fields defined in Figure 267 to the desired values (e.g., taking into account the common namespace capabilities, available capacity);
5. Host software issues the Namespace Management command specifying the Create operation and the data structure. On successful completion of the command, the Namespace Identifier of the new namespace is returned in Dword 0 of the completion queue entry. At this point, the new namespace is not attached to any controller; and
6. Host software requests the Identify Namespace data structure for the new namespace to determine all attributes of the namespace.

...

8.15 Sanitize Operations (Optional)

A sanitize operation alters all user data in the NVM subsystem such that recovery of any previous user data from any cache, the non-volatile media, or any Controller Memory Buffer is not possible. It is implementation specific whether Submission Queues and Completion Queues within a Controller Memory Buffer are altered by a sanitize operation; all other data stored in all Controller Memory Buffers is altered by a sanitize operation. If a portion of the user data was not altered and the sanitize operation completed successfully, then the NVM subsystem shall ensure permanent inaccessibility of that portion of the user data for any future use within the NVM subsystem (e.g., retrieval from NVM media, caches, or any Controller Memory Buffer) and permanent inaccessibility of that portion of the user data via any interface to the NVM subsystem, including management interfaces such as NVMe-MI implementation.

The scope of a sanitize operation is all locations in the NVM subsystem that are able to contain user data, including caches and unallocated or deallocated areas of the media. If the composition of the NVM subsystem (refer to section 7.NEW) changes (e.g., a new domain is added, or a division event occurs) and that change prevents the successful completion of a sanitize operation, then the sanitize operation shall fail. Sanitize operations do not affect the Replay Protected Memory Block, boot partitions, or other media and caches that do not contain user data. A sanitize operation also may alter log pages as necessary (e.g., to prevent derivation of user data from log page information). A sanitize operation is only able to be started if the NVM subsystem is not divided (refer to section 7.NEW). Once a sanitize operation is started, it cannot be aborted and continues after a Controller Level Reset including across power cycles.

The Sanitize command (refer to section 5.24) is used to start a sanitize operation or to recover from a previously failed sanitize operation. All sanitize operations are performed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation). The completion of a sanitize operation is indicated in the Sanitize Status log page, and with the Sanitize Operation Completed asynchronous event (if an Asynchronous Event Request Command is outstanding).

The following information is modified …:

8.17 Endurance Groups

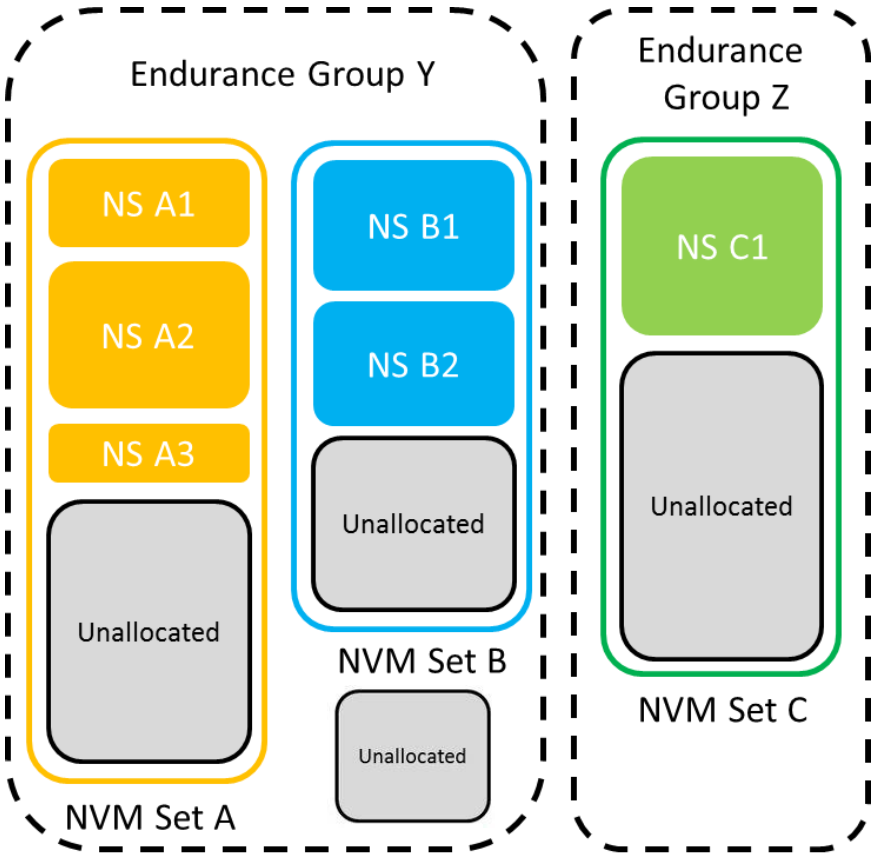
Endurance may be managed within a single NVM Set (refer to section 4.9) or across a collection of NVM Sets. Each NVM Set is associated with an Endurance Group (refer to Figure 250). If two or more NVM Sets have the same Endurance Group Identifier, then endurance is managed by the NVM subsystem across that collection of NVM Sets. If only one NVM Set is associated with a specific Endurance Group Identifier, then endurance is managed locally to that NVM Set. An Endurance Group shall be part of only one domain (refer to section 7.NEW).

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

The ~~endurance~~ information ~~for~~ that describes an Endurance Group is ~~specified~~ indicated in the Endurance Group Information log page (refer to section 5.14.1.9).

Figure 484 shows Endurance Groups added to the example in Figure 133. In this example, the endurance of NVM Set A and NVM Set B are managed together as part of Endurance Group Y, while the endurance of NVM Set C is managed only within NVM Set C which is the only NVM Set that is part of Endurance Group Z.

Figure 484: NVM Sets and Associated Namespaces



…

The following information is modified as shown :

8.20 Asymmetric Namespace Access Reporting (Optional)

…

8.20.1 Asymmetric Namespace Access Reporting Overview

Asymmetric Namespace Access (ANA) occurs in environments where namespace access characteristics (e.g., performance or ability to access the media) may vary based on:

- the controller used to access the namespace (e.g., Fabrics); and
- the internal configuration of the NVM subsystem.

Asymmetric Namespace Access Reporting is used to indicate to the host information about those access characteristics.

…

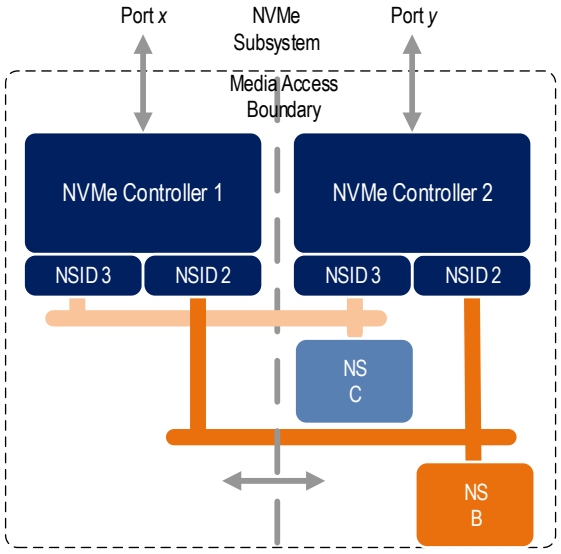
Namespaces attached to a controller that supports Asymmetric Namespace Access Reporting shall:

- be members of an ANA Group; and
- supply a valid ANA Group Identifier in the ANA Group Identifier (ANAGRPID) field in Identify Namespace data structure (refer to Figure 245).

A controller that supports Asymmetric Namespace Access Reporting may also support multiple domains (refer to section 7.NEW).

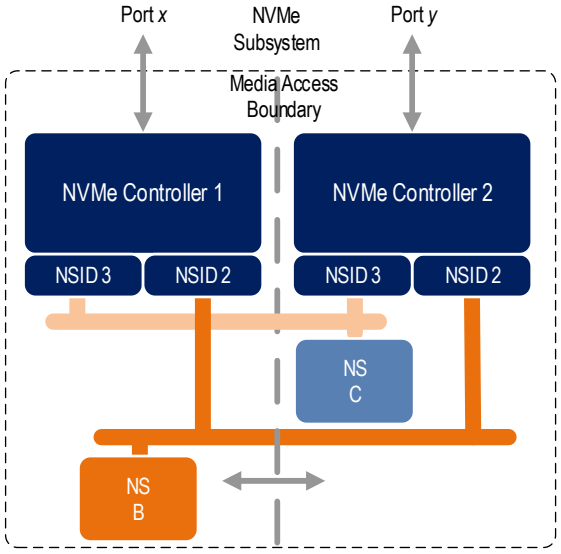
Figure 491 shows an example of an NVM subsystem where access characteristics vary as a result of the presence of two independent domains. In this example, the non-volatile media for namespace B and for namespace C are contained within the same domain that contains controller 2. ~~As a result~~ In this example, controller 2 provides optimized access to namespace B and to namespace C while controller 1 does not provide optimized access to namespace B or to namespace C. In an NVM subsystem that supports multiple domains (refer to section 7.NEW), the Media Access Boundary shown in Figure 491 may be a Communication boundary as shown in Figure 9996 and Figure 9997.

Figure 491: Namespace B and C optimized through Controller 2



To provide optimized access to namespace B through controller 1, the NVM subsystem may be administratively reconfigured, or may perform autonomous internal reconfiguration actions that change the access characteristics of namespace B when accessed through controller 1 and controller 2 as shown in Figure 492. Controller 2 provides optimized access to namespace C while controller 1 provides optimized access to namespace B. In an NVM subsystem that supports multiple domains (refer to section 7.NEW), the Media Access Boundary shown in Figure 492 may be a Communication boundary as shown in Figure 9996 and Figure 9997.

Figure 492: Namespace B optimized through Controller 1



8.20.2 ANA Groups

Namespaces that are members of the same ANA Group perform identical asymmetric namespace access state transitions. The ANA Group maintains the same asymmetric namespace access state for all namespaces that are members of that ANA Group (i.e., a change in the asymmetric namespace access state of one namespace only occurs as part of a change in the asymmetric namespace access state of all namespaces that are members of that ANA Group). Namespaces that are members of the same ANA Group shall be members of the same domain. The method for assigning namespaces to ANA Groups is outside the scope of the specification.

An ANA Group may contain zero or more namespaces, zero or more NVM Sets, or zero or more Endurance Groups. The mapping of namespaces, NVM Sets, and Endurance Groups to ANA Groups is vendor specific.

A valid ANA Group Identifier is a non-zero value that is less than or equal to ANAGRPMAX (refer to Figure 109).

…

8.20.4 Asymmetric Namespace Access States Command Processing Effects

Processing of Admin commands that:

- are not NVM Command Set Specific commands; and
- do not use the Namespace Identifier (i.e., Figure 139 – “Namespace Identifier Used” column indicates “No”),

are not affected by ANA states, except as specified in Figure 494.

Figure 494 describes Asymmetric Namespace Access effects on command processing.

Figure 494: ANA effects on Command Processing

Command	ANA State	Effects on command processing
Get Features	ANA Inaccessible, ANA Persistent Loss, or ANA Change	The following feature identifiers are not available[1]: a) LBA Range Type (i.e., 03h); b) Error Recovery (i.e., 05h); c) Write Atomicity Normal (i.e., 0Ah); d) Reservation Notification Mask (i.e., 82h); and e) Reservation Persistence (i.e., 83h).
Get Log Page	ANA Inaccessible, ANA Persistent Loss, or ANA Change	The following log pages are affected: a) Error Information (i.e., 01h): The log page may contain entries only for namespaces whose relationship to the controller processing the command is in the ANA Optimized state (refer to section 8.20.3.1) or the ANA Non-optimized state (refer to section 8.20.3.2).
Identify	ANA Inaccessible or ANA Persistent Loss	Capacity fields in the Identify namespace data structure (refer to Figure 245) information is cleared to 0h.

Figure 494: ANA effects on Command Processing

Command	ANA State	Effects on command processing
Set Features	ANA Inaccessible	The saving of features shall not be supported and the following feature identifiers are not available[1]: a) LBA Range Type (i.e., 03h); b) Error Recovery (i.e., 05h); c) Write Atomicity Normal (i.e., 0Ah); d) Reservation Notification Mask (i.e., 82h); and e) Reservation Persistence (i.e., 83h). If the NSID is set to FFFFFFFFh, then the command shall fail [2] with a status code of Asymmetric Access Inaccessible (refer to section 8.20.3.3).
	ANA Change	The saving of features shall not be supported and the following feature identifiers are not available[1]: a) LBA Range Type (i.e., 03h); b) Error Recovery (i.e., 05h); c) Write Atomicity Normal (i.e., 0Ah); d) Reservation Notification Mask (i.e., 82h); and e) Reservation Persistence (i.e., 83h). If the NSID is set to FFFFFFFFh, then the command shall fail [2] with a status code of Asymmetric Access Transition (refer to section 8.20.3.5).
	ANA Persistent Loss	This command shall fail [2] with a status code of Asymmetric Access Persistent Loss (refer to section 8.20.3.4).

NOTES:
1. If the ANA state is ANA Inaccessible State, then commands that use feature identifiers that are not available shall fail with a status code of Asymmetric Access Inaccessible. If the ANA state is ANA Persistent Loss State, then commands that use feature identifiers that are not available shall fail with a status code of Asymmetric Access Persistent Loss. If the ANA state is ANA Change State, then commands that use feature identifiers that are not available shall fail with a status code of Asymmetric Access Transition.
2. If any namespace that is attached to the controller is in an ANA Group that is in the ANA Inaccessible state, the ANA Persistent Loss state, or the ANA Change state, then the command shall fail with the indicated status. Depending on the ANA state of the ANA Group that contains a namespace (e.g., an ANA state changes during the processing of the command), the specified feature identifier may be altered for some attached namespaces and not altered for other attached namespaces.

10.5 Controller Fatal Status Condition

…

If the Controller Fatal Status (CSTS.CFS) bit is set to '1' on any controller in the NVM subsystem, the host should issue a Controller Reset to that controller.

If that Controller Reset does not clear the Controller Fatal Status condition, the host should initiate an NVM Subsystem Reset (refer to section 7.3~~.1~~), if supported.

…