NVM Express® Technical Proposal

Technical Proposal ID	TP6011 - NVMe Scalable Resource Management
Change Date	10/26/2022
Builds on Specifications	NVMe Base Specification 2.0a NVMe-MI Specification 1.2

Technical Proposal Author(s)

Name	Company
Phil Cayton	Intel
Rajalaxmi Angadi	Intel
Nick Adams	Intel
Dave Minturn	Intel
Peter Onufryk	Intel
Fred Knight	NetApp
David Black	Dell/EMC
Krishnakumar Gowravaram	Cisco Systems

Prior to Hosts receiving Discovery log pages, prior to Hosts being able to connect to remote (NVMe-oF) storage resources, those resources must be enumerated, configured, and provisioned to Hosts. When an administrator wants to expose NVM resources over fabrics they must statically or manually configure and assign NVM resources (e.g., namespaces) to each exported NVM Subsystem. In addition, some use-cases require setting up individual access rights to each exported NVM Subsystem to provision user-access. Additionally, fabrics connected storage resources require configuring and assigning each exported NVM Subsystem with fabric transports. When an exported NVM Subsystem or its associated resources are no-longer needed, administrators must individually deprovision exported resources so the underlying NVM resources may be recovered and/or sanitized and then go through the process of recovering those resources.

This administrative overhead presents a scalability and flexibility issue for NVMe and NVMe-oF, especially in complex and dynamic installations where the workloads and/or datasets vary, or where infrastructure and systems, fabric devices, and/or storage devices change. Static configurations lack flexibility, individual manual configuration and administration decreases practical scalability.

In addition, a lack of common logical resource creation and management capabilities has led to the proliferation of vendor specific tools which limits interoperability and common configuration mechanisms for exported NVMe resources. This leads to either single-vendor solutions or to having to support and use multiple management applications.

This proposal seeks to add the notion of exported NVM Subsystems and a common framework for NVMe and NVMe-oF to dynamically construct, configure, and provision exported NVM Subsystems as logical aggregations of underlying physical resources. In a landlord/tenant environment, e.g., an Infrastructure-as-a-Service cloud, a landlord is able to export NVM Subsystems to each tenant, thereby providing each tenant with separate (non-interfering) scopes of management, namely her own exported NVM Subsystems. This functionality is intended to support a variety of use-case / transport-specific virtualization capabilities e.g., NVMe over Fabrics, Multi-path I/O and Namespace Sharing, and Access from Virtual Machines.

This Technical Proposal addresses the above issues by defining a common framework for configuration and management of exported NVMe resources via administrative commands by:

1. Introducing terms to disambiguate underlying NVMe resources from exported NVMe resources
2. Defining new optional administrative management operations to enable resource management for logical representations of physical NVM resources including:
- retrieval of information regarding available NVM <u>and</u> fabrics resources on underlying NVM storage for the purpose of configuring exported NVM resources.
- configuration of exported NVM resources
- provisioning access to exported NVM resources

This Technical Proposal **does not**
- Define new security mechanisms
- Affect preconfigured underlying NVM subsystem configuration

Revision Date	Change Description
04/06/2020	Initial draft post Phase 1 exit vote
03/13/2021	Draft reviewed with the FMDS TWG
05/17/2021	Phase 3 Rework post Phase 2 exit vote
06/22/2021	Draft reviewed with the FMDS TWG
06/24/2021	Draft reworded after review with Nick Adams
06/29/2021	Draft reviewed with FMDS TWG
07/15/2021	Changes based on Michael Allison feedback
07/22/2021	Changes based on FMDS TWG
08/16/2021	Changes based on Fred Knight, Michael Allison, and David Black feedback
10/15/2021	Changes based on post-30-day review feedback by Michael Allison, David Black, Fred Knight
12/17/2021	Changes based on post-30-day review feedback by David Black, Fred Knight
01/04/2022	Changes based on Michael Allison feedback
01/06/2022	Changes based on NVMe Technical Work Group discussion
02/01/2022	Editorial changes based on Michael Allison feedback.
02/15/2022	Editorial changes based on 30-day review and FMDS feedback
03/30/2022	Integrated
4/4/2022	Integration comment resolution.
5/2/2022	Additional Integration comment resolution based on Phil Cayton and Fred Knight feedback.
06/30/2022	Address comments received by reviewers, requests for changes regarding Namespace Management/attach, and to address lack of support on controllers using a Memory-Based Transport Models (e.g., PCIe transport).
07/21/2022	Changes based on requests from the NVMe TWG / FMDS prior to reentry to 30-day review.
10/10/2022	Integrated changes based on 30-day review comments.
10/19/2022	Editorial changes based on Michael Allison and Phil Cayton feedback.

Description for Changes Document for NVMe Base Specification 2.0a

New features/feature Enhancements/Required Change:
NVMe and NVMe-oF management of exported NVMe resources (optional)
Modify section 1.5 to add additional "Definitions" to support disambiguate underlying NVMe resources from exported NVMe resources and to support NVMe scalable resource management. Also required is rearranging existing subsections in 1.5 to keep the definitions in alphabetical order.
Notes sanitize, Predictable Latency, and Endurance group commands are prohibited for Exported NVM Resources. Adds section discussing Sanitize operation Effects on Exported NVM Subsystems
Notes capacity Management commands are prohibited for Exported NVM Resources.

Adds new Command Specific Status Values to figure 95 - Status Code – Command Specific Status Values table.

References NVMe Base Specification 2.0a sections: 1.5, 3.1.2, 3.3.3, 5, 8

Splits section 8.11 (Namespace Management) into two subsections - 8.11.TBD1 Namespace Management for NVM Subsystems that are not Exported NVM Subsystems, and 8.11.TBD2 Namespaces Management for NVM Subsystems that are Exported NVM Subsystems to disambiguate Namespace Management depending on the Export/Non-exported state of the NVM Subsystems.

Adds new sections 8.TBD1 – 8.TBD3 to describe exporting NVM resources and show example flows using the added commands and associated details.

2   Markup Conventions:

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

1   Introduction

…

1.5   Definitions

1.5.TBD1   Underlying Namespace

A namespace (defined in section 1.5.36) accessible through physical or virtual functions in an Underlying NVM Subsystem that may be used to associate with an Exported NVM Subsystem. Underlying Namespaces are identified by the Underlying Namespace Entry data structure (refer to Figure TBD2).

1.5.TBD2   Underlying Namespace List

A list of namespaces (refer to section 5.17.2.TBD1) in all underlying NVM subsystems that may be used to create an Exported Namespace

1.5.TBD3   Exported Namespace

A namespace in an Exported NVM Subsystem.

1.5.TBD4   Exported NVM Subsystem

A logical NVM subsystem that exports underlying NVM resources and that:
   a) contains zero or more Exported Namespaces;
   b) contains zero or more controllers;
   c) contains zero or more Exported Ports; and
   d) may contain an Allowed Host List.

1.5.TBD5   Allowed Host List

A list of hosts (identified by Host NQN and Host Identifier) present in each Exported NVM Subsystem that are granted access to the Exported NVM Subsystem via an Exported Port.

1.5.TBD6　　　　Underlying NVM Subsystem

Defined as NVM subsystem.

1.5.TBD7　　　　Underlying Port

A port through which an NVMe subsystem is attached to a transport (e.g., Ethernet, InfiniBand, Fibre Channel) (refer to section 1.5.58).

1.5.TBD8　　　　Ports List

A list of ports that may be used to export an NVM subsystem. Entries in the Ports List are in the format specified by Underlying Fabrics Transport Entry data structure (refer to Figure TBD4).

1.5.TBD9　　　　Exported Port

A port used to export an NVMe subsystem over a specific fabrics transport and represented by an Exported Port ID.

1.5.TBD10　　　　Exported Port ID

A port identifier used to specify an Exported Port.

1.5.TBD11　　　　Exported NVM Resources

NVM resources created to enable remote access to physical NVM resources that includes:
   a)   Exported NVM Subsystems;
   b)   Exported Namespaces; and
   c)   Exported Ports.

Modify section 3.1.2.1.1 "Command Support" to update Figure 22 "IO Controller – Admin Command Support" in order to add commands that are mandatory, optional, and prohibited for an IO Controller.

3   NVM Express Architecture

…

3.1   NVM Controller Architecture

…

3.1.2   Controller Types

…

3.1.2.1   I/O Controller

…

3.1.2.1.1   Command Support

Figure 22 and Figure 23 define commands that are mandatory, optional, and prohibited for an I/O controller. I/O Command Set specific command support requirements are described within individual I/O Command Set specifications.

Figure 22: I/O Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Delete I/O Submission Queue	M	5.7
Create I/O Submission Queue	M	5.5
Get Log Page	M	5.16
…	…	…
Sanitize	O[4]	5.24
…	…	…
Authentication Receive	O[3]	6.1
Disconnect	O[3]	6.4
Clear Exported NVM Resource Configuration	P	5.TBD1
Create Exported NVM Subsystem	P	5.TBD2
Manage Exported NVM Subsystem	P	5.TBD3
Manage Exported Namespace	P	5.TBD4
Manage Exported Port	P	5.TBD5
I/O Command Set Specific Admin Command	Refer to the applicable I/O Command Set specification	Refer to the applicable I/O Command Set specification
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. For NVMe over PCIe implementations, all Fabrics commands are prohibited. For NVMe over Fabrics implementations, the commands are as noted in the table.
4. Prohibited for an Exported NVM Subsystem (refer to section 8.TBD1).

Modify section 3.2.2.1.1 "Command Support" to update Figure 28 "Administrative Controller – Admin Command Support" in order to add commands that are mandatory, optional, and prohibited for an Administrative Controller.

Editor's note *– several entries in Figure 28, "Administrative Controller – Admin Command Support" prior to this TP reference "note 6" but there seems to be no "note 6". Hence the addition of "note_TBD" rather than a new note #.*

3.1.2.2   Administrative Controller

…

3.1.2.2.1 Command Support

Figure 28 defines 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 28: Administrative Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Delete I/O Submission Queue	P	5.7

Technical input submitted to NVM Express® is subject to the terms of the NVM Express® Participant's agreement. Copyright © 2008 to 2022 NVM Express, Inc.

Figure 28: Administrative Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Create I/O Submission Queue	P	5.5
Get Log Page	M	5.16
…	…	…
Sanitize	O<mark>Note_TBD</mark>	5.24
…	…	…
Authentication Receive	O[6]	6.1
Disconnect	O[6]	6.4
Clear Exported NVM Resource Configuration	O	5.TBD1
Create Exported NVM Subsystem	O	5.TBD2
Manage Exported NVM Subsystem	O	5.TBD3
Manage Exported Namespace	O	5.TBD4
Manage Exported Port	O	5.TBD5
I/O Command Set Specific Admin Commands	P	
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 30 are implemented.
4. Mandatory if Telemetry Log, Firmware Commit, or SMART/Health Critical Warnings are supported.
NOTE_TBD Prohibited for an Exported NVM Subsystem (refer to section 8.TBD1).
For NVMe over PCIe implementations, all Fabrics commands are prohibited. For NVMe over Fabrics implementations, the commands are as noted in the table.

Modify section 3.1.2.2.2 "Log Page Support" to update Figure 29 "Administrative Controller – Log Page Support" to add comment '5' to note which Log Pages are Prohibited to be supported for an Exported NVM Subsystem.

3.1.2.2.2 Log Page Support

Figure 29 defines log pages that are mandatory, optional, and prohibited for an Administrative controller.

Figure 29: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements [1]
…	…
Endurance Group Information	O[5]
Predictable Latency Per NVM Set	O[5]
Predictable Latency Event Aggregate	O[5]
…	…

Figure 29: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements [1]
Sanitize Status	O[5]
Notes:	
1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited	
2. Optional if set Features command is not supported (refer to Figure 28).	
3. Optional if NVMe-MI Send command and NVMe-MI Receive command is not supported (refer to Figure 28).	
4. Optional for NVM Express revision 1.4 and earlier.	
5. Prohibited for an Exported NVM Subsystem (refer to section 8.TBD1).	

Modify section 3.1.2.2.3 "Features Support" to update Figure 30 "Administrative Controller – Feature Support to add comment '7' to note features are prohibited for Exported NVM Subsystems.

3.1.2.2.3 Features Support

Figure 30 defines features that are mandatory, optional, and prohibited for an Administrative controller.

Figure 30: Administrative Controller – Feature Support

Feature Name	Feature Support Requirements	Logged in Persistent Event Log[1]
Arbitration	P	P
Power Management	O	NR
…	…	…
Read Recovery Level Config	O	O
Predictable Latency Mode Config	O[7]	P
Predictable Latency Mode Window	O[7]	O[7]
Host Behavior Support	O	O
Sanitize Config	O[7]	O[7]
Endurance Group Event Configuration	O[7]	O[7]
…	…	…
Rotational Media	P	P
Notes:		
1. O/M/P/NR definition: O = Optional, M = Mandatory, P = Prohibited, NR = Not Recommended.		
2. The feature is mandatory 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.		
4. Mandatory if reservations are supported as indicated in the Identify Controller data structure.		
5. 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.		
6. This feature is optional for NVM subsystems that do not implement a Management Endpoint. For NVM subsystems that implement any Management Endpoint refer to the NVMe Management Interface Specification.		
7. Prohibited for an Exported NVM Subsystem (refer to section 8.TBD1).		

Modify section 3.1.2.3.2 "Command Support" to update Figure 32 "Discovery Controller – Admin Command Support" in order to add commands that are mandatory, optional, and prohibited for a Discovery Controller.

3.1.2.3 Discovery Controller

…

3.1.2.3.2 Command Support

A Discovery controller supports all mandatory Fabrics commands. A Discovery controller supports a subset of Admin commands shown in figure 32.

Figure 32: Discovery Controller – Admin Command Support

Command	Command Support Requirements [1]	Reference
Get Log Page	M	5.16
Delete I/O Completion Queue	P	5.6
Create I/O Completion Queue	P	5.4
…	…	…
Authentication Receive	O	6.1
Disconnect	O	6.4
Clear Exported NVM Resource Configuration	P	5.TBD1
Create Exported NVM Subsystem	P	5.TBD2
Manage Exported NVM Subsystem	P	5.TBD3
Manage Exported Namespace	P	5.TBD4
Manage Exported Port	P	5.TBD5
I/O Command Set Specific Admin Commands	P	
Vendor Specific	O	
Notes:		
1. O = Optional, M = Mandatory, P = Prohibited		
2. For Discovery controllers that do not support explicit persistent connections, the command is reserved. For Discovery controllers that support explicit persistent connections, the command is mandatory.		

Modify section 3.3.3.2.1.2 to update Figure 95 "Status Code – Command Specific Status Definition" in order add new Command Specific Status Values

3   NVM Express Architecture

…

3.3   NVM Queue Models

…

3.3.3   Queueing Data Structures

…

3.3.3.2   Common Completion Queue Entry

…

3.3.3.2.1   Status Field Definition

…

3.3.3.2.1.2 Command Specific Status Definition

Figure 95: Status Code – Command Specific Status Values

Value	Description	Commands Affected
00h	Completion Queue Invalid	Create I/O Submission Queue
01b	Invalid Queue Identifier	Create I/O Submission Queue, Create I/O Completion Queue, Delete I/O Completion Queue, Delete I/O Submission Queue
…	…	…

34h	ZoneGroup Originator Invalid	Fabric Zoning Send
35h	Invalid Host	Manage Exported NVM Subsystem
36h	Invalid NVM Subsystem	Manage Exported NVM Subsystem
37h to 6Fh	Reserved	
…	…	…

Modify section 5 to update Figure 138 "Opcodes for Admin Commands" in order add new commands and associated details.

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 3.3.3. The Completion Queue Entry (CQE) structure and the fields that are common to all Admin commands are defined in section 3.3.3.2. The command specific fields in the SQE and CQE structures (i.e., SQE Command Dwords 10 to 15, CQE Dword 0, and CQE Dword 1) for the Admin Command Set are defined in this section.

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 138 defines all Admin commands. Refer to Figure 22, Figure 28, and Figure 32 for mandatory, optional, and prohibited commands for the various controller types.

Figure 138: Opcodes for Admin Commands

Opcode by Field			Combined Opcode[1]	Namespace Identifier Used[2]	Command	Command Set Specific[8]
(07) Generic Command	(06:02) Function	(01:00) Data Transfer[3]				
0b	000 00b	00b	00h	No	Delete I/O Submission Queue	No
0b	000 00b	01b	01h	No	Create I/O Submission Queue	No
…	…	…	…	…	…	…
0b	001 11b	01b	1Dh	No	NVMe-MI Send	No
0b	001 11b	10b	1Eh	No	NVMe-MI Receive	No
0b	010 10b	00b	28h	No	Clear Exported NVM Resource Configuration[10]	No
0b	010 10b	10b	2Ah	No	Create Exported NVM Subsystem[10]	No
0b	010 11b	01b	2Dh	No	Manage Exported NVM Subsystem[10]	No
0b	011 00b	01b	31h	Yes	Manage Exported Namespace[10]	No
0b	011 01b	01b	35h	No	Manage Exported Port[10]	No
0b	010 00b	00b	20h	No	Capacity Management	No
…	..	…	…	**…**	…	…

NOTES:
1. Opcodes not listed are reserved.
2. A subset of commands use the Namespace Identifier (NSID) field. If the Namespace Identifier field is used, then the value FFFFFFFFh is supported in this field unless otherwise indicated in footnotes in this figure that a specific command does not support that value or supports that value only under specific conditions. When this field is not used, the field is cleared to 0h as described in Figure 87.
3. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
4. This command does not support the use of the Namespace Identifier (NSID) field set to FFFFFFFFh.
5. Support for the Namespace Identifier field set to FFFFFFFFh depends on the Directive Operation (refer to section 8.7).
6. Use of the Namespace Identifier field depends on the CNS value in the Identify Command as described in Figure 273.

7. The use of the Namespace Identifier is Security Protocol specific.
8. No = Not I/O Command Set specific, A = All I/O Command Sets, NVM = NVM Command Set specific, ZNS = Zoned Namespace Command set.
9. All Fabrics commands use the opcode 7Fh. Refer to section 6 for details.
10. Support for this command is prohibited in NVM subsystems that use a Memory-Based Transport Model (e.g., the PCIe transport) for any controller.

Editor's Note: Since commands in this TP are prohibited during sanitize operations, no changes to Figure 139 (Sanitize Operations and Format NVM Command – Admin Commands Allowed) are necessary.

Modify section 5.17.1 to modify figure 273 and update the Identify Command to add a new CNS to support the "Get Underlying Namespace List" and "Get Ports List" operations

5.17   Identify command

…

5.17.1 Identify Command Overview

Figure 273: Identify – CNS Values

CNS Value	O/M[1]	Definition	NSID[2]	CNTID[3]	CSI[4]	Reference Section
		Active Namespace Management				
…		…	…	…	…	…
		Controller and Namespace Management				
…	…	…	…	…	…	…
1Ch	O	I/O Command Set data structure	N	Y	N	5.23.2.21
1Dh	O	Get Underlying Namespace List[Note_TBD]	Y	Y	N	5.17.2.TBD1
1Eh	O	Get Ports List	N	N	N	5.17.2.TBD2
to 1Fh		Reserved				
		Future Definition				
20h to FFh		Reserved				
	Notes: … Note_TBD. Support for this CNS value is prohibited in NVM subsystems that use a Memory-Based Transport Model (e.g., the PCIe transport) for any controller.					

Add new sections 5.17.2.TBD1 and 5.17.2.TBD2 to add to add new CNS values to support NVMe and NVMe-oF Scale out management

5.17.2   Identify Data Structures

…

5.17.2.TBD1 Get Underlying Namespace List (CNS 1Dh)

An Underlying Namespace List data structure (refer to Figure TBD1) is returned containing a list of all Underlying Namespaces spanning all NVM subsystems that are accessible through either a virtual function or a physical function.

Figure TBD1: Underlying Namespace List Data Structure

Bytes	Description
07:00	**Generation Counter (GENCTR):** This field indicates the version of Underlying Namespace information. This field shall be cleared to 0h as a result of NVM Subsystem Reset of the Underlying NVM Subsystem. For each change in the Underlying Namespace List, this counter is incremented by one. If the value of this field is FFFFFFFF_FFFFFFFFh, then the field shall be cleared to 0h when incremented (i.e., rolls over to 0h).
15:08	**Number Entries (N):** Number of Underlying Namespace Entries.
Underlying Namespace List	
335:16	**Underlying Namespace Entry 1:** The first Underlying Namespace Entry data structure (refer to Figure TBD2) in the Underlying Namespace List. If any.
655:336	**Underlying Namespace Entry 2:** If any.
…	…
320*N+15:320*(N-1)+16	**Underlying Namespace Entry N:** If any.

Figure TBD2: Underlying Namespace Entry Data Structure

Bytes	Description
255:00	**Underlying NVM Subsystem NQN:** The Underlying NVM Subsystem NQN which contains the Underlying Namespace.
259:256	**Namespace Identifier (NSID):** This field indicates the NSID value of the namespace in the Underlying NVM Subsystem identified by the Underlying NVM Subsystem NQN field. If the value in this field is not an NSID reported in the Underlying Namespace List (refer to 5.17.2.TBD1), then the command shall be aborted with a status code set to Invalid Field in Command.
261:260	**Controller ID (CNTLID):** Contains the Underlying NVM Subsystem unique controller identifier associated with the Underlying Namespace. Refer to section 5.17.2.1.
319:262	Reserved

5.17.2.TBD2 Get Ports List (CNS 1Eh)

A Ports List data structure (refer to Figure TBD3) is returned containing a list of Underlying Ports that may be used to export NVMe over Fabrics NVM subsystems.

Figure TBD3: Ports List Data Structure

Bytes	Description
07:00	**Generation Counter (GENCTR):** This field indicates the version of Port information. This field shall be cleared to 0h as a result of NVM Subsystem Reset of the Underlying NVM Subsystem. For each change in the Ports List, this counter is incremented by one. If the value of this field is FFFFFFFF_FFFFFFFFh, then the field shall be cleared to 0h when incremented (i.e., rolls over to 0h).
15:08	**Number Entries (N):** Number of Fabrics Transport Entries in the Ports List.
Ports List	
591:16	**Fabrics Transport Entry 1: T**he first Underlying Fabrics Transport Entry data structure (refer to Figure TBD4) in the Ports List, if any.
1167:592	**Fabrics Transport Entry 2**: If any.
…	…
576*N+15:576*(N-1)+16	**Fabrics Transport Entry N**: If any.

Figure TBD4: Underlying Fabrics Transport Entry Data Structure

Bytes	Description
255:00	**Transport Address (TRADDR):** This field indicates the address of the NVM subsystem that may be used for a Connect command as an ASCII string. The Transport Address Family field describes the reference for parsing this field. Refer to section 1.4 for ASCII string requirements. For the definition of this field, refer to the appropriate NVMe Transport specification.
511:256	**Transport Specific Address Subtype (TSAS):** This field indicates NVMe Transport specific information about the address. For the definition of this field, refer to the appropriate NVMe Transport specification.
513:512	**Port ID of the Underlying Port** (refer to Figure 264)
514	**Transport Type (TRTYPE):** Specifies the NVMe Transport Type. Refer to Figure 264.
515	**Transport Address Family (ADRFAM):** Specifies the Address Family. Refer to Figure 264.
516	**Transport Requirements (TREQ):** This field indicates requirements for the NVMe Transport. Refer to Figure 264.
575:517	Reserved

Modify section 5.22 to modify text about Namespace Attachment command

5.22 Namespace Attachment command

The Namespace Attachment command is used to attach and detach controllers from a namespace (refer to section 8.11). The attach and detach operations are persistent across all reset events. Namespace attach and detach operations are persistent across Virtualization Management commands that set a secondary controller offline.

~~If the Namespace Attachment command is supported, then the Namespace Management command (refer to section 5.23) shall also be supported.~~

The Namespace Attachment command uses the Data Pointer and Command Dword 10 fields. All other command specific fields are reserved.

[…]

Modify section 5.23 to modify text about Namespace Management command support

5.23 Namespace Management command

The Namespace Management command is used to manage namespaces (refer to section 8.11), including create and delete operations.

Note: The controller continues to execute commands submitted to I/O Submission Queues while this operation is in progress.

If the Namespace Management command is supported, then the Namespace Attachment command (refer to section 5.22) shall also be supported. The Namespace Management command shall not be supported by controllers in an Exported NVM subsystem.

Host software uses the Namespace Attachment command to attach or detach a namespace to or from a controller. The create operation does not attach the namespace to a controller. As a side effect of the delete operation, the namespace is detached from all controllers as the namespace is no longer present in the system. It is recommended that host software detach all controllers from a namespace prior to deleting the namespace. If the namespace is attached to another controller (i.e., a controller other than the controller processing the operation) and that controller has Namespace Attribute Notices enabled (refer to Figure

326), when a delete operation is requested, then as part of the delete operation a Namespace Attribute Notice is issued by that controller to indicate a namespace change.

The data structure used for the create operation is defined by the I/O Command Set specified in the CSI field (refer to the applicable I/O Command Set specification). There is no data structure transferred for the delete operation.

[…]

Modify section 5.24 to add a new paragraph that prohibits the Sanitize command for Exported NVM subsystems.

5.24 Sanitize Command

[…]

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 3.2.4), then the Sanitize command shall be aborted with a status code of Asymmetric Access Inaccessible or Asymmetric Access Persistent Loss.

The Sanitize command shall not be supported by Exported NVM Subsystems (refer to section 8.21.TBD1).

[…]

Add new sections 5.TBD1 – 5.TBD5 (and associated subsections) to add new commands and associated details.

5.TBD1 Clear Exported NVM Resource Configuration command

The Clear Exported NVM Resource Configuration command is used to delete all Exported NVM resource configuration information including removing all Exported NVM Resources (refer to section 1.5.TBD10 for a list of Exported NVM Resources).

All commands specific fields are reserved.

The Clear Exported NVM Resource Configuration command shall not be supported by Exported NVM Subsystems.

The Clear Exported NVM Resource Configuration command shall not affect Underlying Namespaces or Underlying NVM Subsystems.

If a Clear Exported NVM Resource Configuration command is issued without first disconnecting all hosts from all Exported NVM Subsystems, then the controller shall abort the command with a status code of Command Sequence Error.

5.TBD1.1 Command Completion

Upon completion of the Clear Exported NVM Resource Configuration command, the controller shall post a completion queue entry to the Admin Completion Queue indicating the status for the command. Refer to section 8.TBD1 for usages.

5.TBD2 Create Exported NVM Subsystem command

The Create Exported NVM Subsystem command is used to create a new Exported NVM Subsystem. The Create Exported NVM Subsystem command uses the Data Pointer and Command Dword 10. All other command specific fields are reserved.

The Create Exported NVM Subsystem command shall not be supported by Exported NVM Subsystems.

Figure TBD5: Create Exported NVM Subsystem – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure TBD6: Create Exported NVM Subsystem – Command Dword 10

Bits	Description
31:09	Reserved
08	**Restricted Access:** A value of '0' configures this Exported NVM Subsystem for unrestricted access and may be accessed by any host. A value of '1' configures this Exported NVM Subsystem to restrict access to Host NQNs present in this Exported NVM Subsystem's Allowed Host List.
07:00	Reserved

The Restricted Access bit defined in Figure TBD6 specifies the initial access setting for the Exported NVM Subsystem.

5.TBD2.1 Command Completion

Upon successful completion of the Create Exported NVM Subsystem command:

- a new Exported NVM Subsystem is created with the Access Control mode specified in the Restricted Access bit of the command:
    a) the newly created Exported NVM Subsystem shall have an empty list of allowed Host NQNs; and
    b) there are no Exported Namespaces linked to the Exported NVM Subsystem;
- the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command (refer to section 8.TBD1 for usages); and
- the SUBNQN for the newly created Exported NVM Subsystem shall be returned in the data buffer.

5.TBD3 Manage Exported NVM Subsystem command

The Manage Exported NVM Subsystem command is used to configure and manage an Exported NVM Subsystem.

The Manage Exported NVM Subsystem command uses the Data Pointer and Command Dword 10. All other command specific fields are reserved.

The Select field defined in Figure TBD8 determines which management operation is to be performed by this command. The specified management operation determines the data structure used as part of the command. The data structure is 4,096 bytes in size (refer to section 5.TBD3.1 for a description of each management operation).

The Manage Exported NVM Subsystem command shall not be supported by Exported NVM Subsystems.

Figure TBD7: Manage Exported NVM Subsystem – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure TBD8: Manage Exported NVM Subsystem – Command Dword 10

Bits	Description		
31:16	Reserved		
15:08	**Management Operation Specific:** If not defined for the management operation specified by the Select field, this field is reserved.		
07:00	**Select (SEL):** This field selects the type of management operation to perform.		
	Management Operation	Description	Reference Section
	00h	Reserved	
	01h	Delete	5.TBD3.1.1
	02h	Change Access Mode	5.TBD3.1.2
	03h	Grant Host Access	5.TBD3.1.3
	04h	Revoke Host Access	5.TBD3.1.4
	05h to FFh	Reserved	

5.TBD3.1 Manage Exported NVM Subsystem Management Operations

5.TBD3.1.1 Delete (Management Operation 01h)

The Delete operation of the Manage Exported NVM Subsystem command is used to delete a specified Exported NVM Subsystem. The data buffer for the Delete operation of the Manage Exported NVM Subsystem command contains an NVM Subsystem NQN specifying the Exported NVM Subsystem to be deleted.

The Management Operation Specific field in Command Dword 10 is reserved and not used by this operation.

If a Manage Exported NVM Subsystem command is processed that specifies the Delete operation without specifying an existing Exported NVM Subsystem identified by the NVM Subsystem NQN (SUBNQN) in the data buffer, then the controller shall abort the command with a status code of Invalid Field in Command.

If there are:
- any active controllers in the specified Exported NVM Subsystem;
- any associations of Exported Namespaces in the specified Exported NVM Subsystem; or
- any Exported Ports exist in the specified Exported NVM Subsystem,

then the controller shall abort the command with a status code of Command Sequence Error.

Upon successful completion of a Manage Exported NVM Subsystem command with a Delete operation the Exported NVM Subsystem identified shall be deleted.

5.TBD3.1.2 Change Access Mode (Management Operation 02h)

The Change Access Mode operation of the Manage Exported NVM Subsystem command is used to change the access mode of an Exported NVM Subsystem to:
- allow all hosts access to the specified Exported NVM Subsystem, or
- restrict access to specified hosts listed in the Allowed Host List associated to the specified Exported NVM Subsystem.

The data buffer for the Change Access Mode operation of the Manage Exported NVM Subsystem command specifies an NVM Subsystem NQN indicating the Exported NVM Subsystem for which the Access Mode is to be changed. The Management Operation Specific field in Command Dword 10 for the Change Access Mode operation is shown in Figure TBD9.

Figure TBD9: Management Operation Specific: Change Access Mode operation

Bits	Description
15:09	Reserved
08	**Restricted Access:** A value of '0' configures this Exported NVM Subsystem for unrestricted access and may be accessed by any host. A value of '1' configures this Exported NVM Subsystem to restrict access to Host NQNs present in the Allowed Host List.

The Restricted Access bit defined in Figure TBD9 specifies the access setting for the Exported NVM Subsystem:
- A value of '0' in the Restricted Access bit configures this Exported NVM Subsystem with Unrestricted Access. This value shall enable all hosts to access the specified Exported NVM Subsystem.

- A value of '1' in the Restricted Access bit shall restrict access of Exported NVM Subsystem to entries in the Allowed Host List associated to the specified Exported NVM Subsystem. Any connected host not in the Allowed Host List associated to the specified Exported NVM Subsystem shall be disconnected from all Exported Namespaces in the Exported NVM Subsystem.

If a Manage Exported NVM Subsystem command is processed that specifies the Change Access Mode operation without specifying an existing Exported NVM Subsystem in the NVM Subsystem NQN (SUBNQN) field, then the controller shall abort the command with a status code of Invalid Field in Command.

Upon successful completion of a Manage Exported NVM Subsystem command with a Change Access Mode operation, the access mode for the specified Exported NVM Subsystem shall be set to the value specified in the Restricted Access bit in the Management Operation Specific field.

5.TBD3.1.3 Grant Host Access (Management Operation 03h)

The Grant Host Access operation of the Manage Exported NVM Subsystem command is used to grant hosts access to Exported NVM Subsystems by adding the specified hosts to the Allowed Host List associated to the specified Exported NVM Subsystem. The data pointer shall point to a Subsystem Management data structure (refer to Figure TBD10). The Management Operation Specific field in Command Dword 10 is reserved and not used by this operation.

The Grant Host Access operation may be used to grant access from a list of hosts (refer to the Host Entry List and "M" in Figure TBD10) to a list of Exported NVM Subsystems (refer to the Exported NVM Subsystem Entry List and "N" in Figure TBD10) through specified port(s) (refer to Port ID of the Underlying Port in Figure TBD12) on each of the Exported NVM Subsystems.

Figure TBD10: Subsystem Management Data Structure

Bytes	Description
63:00	Reserved
65:64	**Number of Host Entries (M)**: Specifies the number of Host Entries in this data structure. This field shall be greater than 0h.
67:66	**Number of Exported NVM Subsystem Entries (N)**: Specifies the number of Exported NVM Subsystem Entries in this data structure. This field shall be greater than 0h.
255:68	Reserved
Host Entry List	
575:256	**Host Entry 1:** Contains a Host Entry data structure as defined by Figure TBD11.
895:576	**Host Entry 2**: Contains a Host Entry data structure as defined by Figure TBD11, if any.
…	…
320*M+255:320*(M-1)+256	**Host entry M:** Contains a Host Entry data structure as defined by Figure TBD11, if any.
Exported NVM Subsystem Entry List	
320+(320*M+255): 320*(N-1)+(320*M+256)	**Exported NVM Subsystem Entry 1:** Contains an Exported NVM Subsystem Entry data structure as defined by Figure TBD12.
…	…
320*N+(320*M+255): 320*(N-1)+(320*M+256)	**Exported NVM Subsystem Entry N:** Contains an Exported NVM Subsystem Entry data structure as defined by Figure TBD12, if any.

Figure TBD11: Host Entry Data Structure

Bytes	Description
07:00	Reserved
23:08	**Host Identifier (HOSTID):** This field has the same definition as the Host Identifier defined in figure 366.
279:24	**Host NVMe Qualified Name (HOSTNQN):** NVMe Qualified Name (NQN) that uniquely identifies the host. Refer to section 4.5.
319:280	Reserved

Figure TBD12: Exported NVM Subsystem Entry Data Structure

Bytes	Description
23:00	Reserved
279:24	**NVM Subsystem NVMe Qualified Name (SUBNQN):** Uniquely identifies an Exported NVM Subsystem. The Allowed Host List associated to this Exported NVM Subsystem is being modified. Refer to section 4.5.
281:280	**Port ID of the Underlying Port** (refer to Figure 264)
319:282	Reserved

The controller shall abort the command with a status code of Invalid Field in Command if:
- the Number of Host Entries field in figure TBD10 is cleared to 0h; or
- the Number of Exported NVM Subsystem Entries field in figure TBD10 is cleared to 0h.

Command specific status values associated with the Grant Host Access operation of the Manage Exported NVM Subsystem command are defined in Figure TBD13. For failures, the byte offset of a failing entry shall be reported in the Command Specific Information field of the Error Information Log Entry.

Figure TBD13: Modify Host Access – Command Operation Specific Status Values

Value	Description
35h	**Invalid Host**
36h	**Invalid NVM Subsystem**

5.TBD3.1.4 Revoke Host Access (Management Operation 04h)

The Revoke Host Access operation of the Manage Exported NVM Subsystem command is used to revoke access to the specified Exported NVM Subsystems from a list of specified hosts. The data pointer shall point to a Subsystem Management data structure (refer to Figure TBD10). The Management Operation Specific field in Command Dword 10 is reserved and not used by this operation.

The Revoke Host Access operation may be used to revoke 1 to N Hosts access from:
- a specified Exported NVM Subsystem through 1 to M specified ports; or
- N different Exported NVM Subsystems through a specified port on each Exported NVM Subsystem.

Connected Hosts that have access revoked from a specified Exported NVM Subsystem on a specified Exported Port shall be disconnected from the specified Exported NVM Subsystem(s) on the specified Exported port.

Command specific status values associated with the Revoke Host Access operation of the Manage Exported NVM Subsystem command are defined in Figure TBD13. For failures, the byte offset of a failing entry shall be reported in the Command Specific Information field of the Error Information Log Entry.

5.TBD3.2 Command Completion

Upon completion of the Manage Exported NVM Subsystem command, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command. Reference sections 5.TBD3.1.1 through 5.TBD3.1.4 for operation specific command completion details. Refer to section 8.TBD1 for usages.

5.TBD4 Manage Exported Namespace command

The Manage Exported Namespace command is used to manage associations of Exported Namespaces with Exported NVM Subsystems. The Manage Exported Namespace command uses the Data Pointer and Command Dword 10. All other command specific fields are reserved.

The Select field defined in Figure TBD15 determines which management operation is to be performed in this command. The specified management operation determines the data structure used as part of the command. The data structure is 4,096 bytes in size (refer to subsections below for description of operation specific data structures).

The Manage Exported Namespace command shall not be supported by Exported NVM Subsystems.

Figure TBD14: Manage Exported Namespace – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure TBD15: Manage Exported Namespace – Command Dword 10

Bits	Description		
31:08	Reserved		
07:00	**Select (SEL):** This field selects the type of management operation to perform.		
	Management Operation	Description	Reference Section
	00h	Reserved	
	01h	Associate Namespace	5.TBD4.1.1
	02h	Disassociate Namespace	5.TBD4.1.2
	03h to FFh	Reserved	

5.TBD4.1 Manage Exported Namespace Management Operations

5.TBD4.1.1 Associate Namespace (Management Operation 01h)

The Associate Namespace operation of the Manage Exported Namespace command is used to create an association between an Exported Namespace ID (ENSID) and an Underlying Namespace; this command also associates the ENSID with an Exported NVM Subsystem identified by the Exported NVM Subsystem NQN.

The Underlying Namespace to be associated with the Exported Namespace ID is returned in the Underlying Namespace List as an Underlying Namespace List Entry (refer to Figure TBD2) and identified with an Underlying Controller ID, Underlying NVM Subsystem, and Underlying Namespace ID combination. That information is specified in an Associate Namespace data structure (refer to Figure TBD16).

The Manage Exported Namespace command with an Associate Namespace operation does not attach the Exported Namespace to a controller. The Exported Namespace becomes an active namespace when attached to a controller due to the processing of a Namespace Attachment command (refer to section 5.22).

The data pointer shall point to an Associate Namespace data structure (refer to Figure TBD16).

Figure TBD16: Associate Namespace Data Structure

Bytes	Description
31:00	**Exported Namespace ID (ENSID)**: A valid and unassociated Exported NSID to be associated to an Underlying Namespace and associated with the specified Exported NVM Subsystem.
253:32	**Exported NVM Subsystem NQN:** Identifies an existing Exported NVM Subsystem to be associated with the Exported Namespace ID.
285:254	**Underlying Namespace ID**: An active Namespace ID to which Exported Namespace ID (ENSID) is to be associated. Refer to Figure 70 for the definition of NSID types and relationship to namespaces.
287:286	**Underlying Controller ID**: Identifies the Underlying Controller used to access the specified Underlying Namespace.
543:288	**Underlying NVM Subsystem NQN**: Identifies the Underlying NVM Subsystem which provides the Underlying Namespace to be associated with the Exported Namespace ID.
575:544	Reserved

The controller shall abort the command with a status code of Invalid Field in Command if:
- a valid Exported Namespace ID is not specified in the Exported Namespace ID field in Figure TBD16;
- the Exported NVM Subsystem NQN field in figure TBD16 does not refer to an existing Exported NVM Subsystem;
- the Underlying Namespace ID field in figure TBD16 does not refer to an active Underlying Namespace ID;
- the Underlying Controller ID field in figure TBD16 does not identify a controller contained in the Underlying NVM Subsystem specified by the Underlying NVM Subsystem NQN field;
- the controller specified by the Underlying Controller ID in the Underlying Controller ID field in figure TBD16 is not attached to the Underlying Namespace specified by the Underlying Namespace ID field specified in figure TBD16;
- the specified Underlying NVM Subsystem NQN field in figure TBD16 does not refer to an existing Underlying NVM Subsystem; or
- the Underlying NVM Subsystem NQN field in Figure TBD16 does not provide the specified Underlying Namespace.

Upon successful completion of a Manage Exported Namespace command with an Associate Namespace operation:

- the NSID which is associated with an active Underlying Namespace is an allocated namespace to the Exported NVM Subsystem. The Exported Namespace and the Underlying Namespace contain the same user data (e.g., format, read, and write operations on the Exported Namespace affect user data stored in the Underlying Namespace; similarly, format, read and write operations on the Underlying Namespace affect user data stored in the Exported Namespace); and
- a Namespace Attribute Changed asynchronous event is reported as described in Figure 146.

5.TBD4.1.2 Disassociate Namespace (Management Operation 02h)

The Disassociate Namespace operation of the Manage Exported Namespace command is used to remove an association between an Exported Namespace ID (ENSID) and an Exported NVM Subsystem and delete the Exported Namespace ID.

The data pointer shall point to a Disassociate Namespace data structure (refer to Figure TBD17).

Figure TBD17: Disassociate Namespace Data Structure

Bytes	Description
31:00	**Exported Namespace ID (ENSID)**: A specified Exported NSID associated with the specified Exported NVM Subsystem
287:32	**Exported NVM Subsystem NQN**: Specifies the Exported NVM Subsystem NQN that is associated with the Exported Namespace ID specified in the ENSID field.
319:288	Reserved

If the Exported Namespace is attached to any controller, then the controller processing the command shall abort the command with a status code of Command Sequence Error.

If the Disassociate Namespace data structure does not contain an Exported NVM Subsystem NQN (SUBNQN) for an existing Exported NVM Subsystem associated with the specified Exported Namespace ID and an Exported NSID associated with the specified Exported NVM Subsystem, then the controller shall abort the command with a status code of Invalid Field in Command.

Upon successful completion of a Manage Exported Namespace command with a Disassociate Namespace operation:

- the specified Exported Namespace ID has been deleted; and
- a Namespace Attribute Changed asynchronous event is reported as described in Figure 146.

5.TBD4.2 Command Completion

Upon completion of the Manage Exported Namespace command, the controller posts a completion queue entry to the Admin Completion queue indicating the status of the command. Refer to section 8.TBD1 for usages.

5.TBD5 Manage Exported Port command

The Manage Exported Port command is used to manage associations of Exported Ports with Exported NVM Subsystems. The Manage Exported Port command uses the Data Pointer and Command Dword 10. All other command specific fields are reserved.

The Select field defined in Figure TBD19 determines which management operation is to be performed in this command. The specified management operation determines the data structure used as part of the command. The data structure is 4,096 bytes in size.

The Manage Exported Port command shall not be supported by Exported NVM Subsystems.

Figure TBD18: Manage Exported Port – Data Pointer

Bits	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure 87 for the definition of this field.

Figure TBD19: Manage Exported Port Data Structure– Command Dword 10

Bits	Description		
31:16	Reserved		
15:08	**Management Operation Specific:** If not defined for the management operation specified by the Select field, this field is reserved.		
07:00	**Select (SEL):** This field selects the type of management operation to perform.		
	Management Operation	Description	Reference Section
	00b	Reserved	
	01h	Create	5.TBD5.1.1
	02h	Delete	5.TBD5.1.2
	04h to FFh	Reserved	

5.TBD5.1 Manage Exported Port Management Operations

5.TBD5.1.1 Create (Management Operation 01h)

The Create operation of the Manage Exported Port command is used to create an Exported Port in an Exported NVM Subsystem and associate the Exported Port with an Underlying Port in the Ports List. The resulting association enables the specified Exported NVM Subsystem to be accessed by hosts through the specified Exported Port. The Exported Port ID for the Exported Port that is created may be specified by the host or generated by the controller.

The data pointer shall point to an Create data structure (refer to Figure TBD21).

Figure TBD20: Management Operation Specific: Create operation

Bits	Description
15:08	Reserved
00	**Generate Exported Port ID:** A value of '0' indicates an Exported Port ID is specified in the data buffer and shall be used to create the new Exported Port. A value of '1' specifies a Port ID shall be generated by the controller and used to identify the Exported Port.

Figure TBD21: Create Data Structure

Bytes	Description
255:00	**Exported NVM Subsystem NVMe Qualified Name (SUBNQN):** NVMe Qualified Name (NQN) that uniquely identifies the Exported NVM subsystem which shall be associated with the created Exported Port. Refer to section 4.5.
257:256	**Exported Port ID (EPID):** If the Generate Exported Port ID bit in the Create operation is cleared to '0', then this field specifies a particular NVM subsystem port to be used for this Exported Transport and associated with the Exported NVM Subsystem identified in the NVM Subsystem NVMe Qualified Name field in this data structure. Different NVMe Transports or address families may utilize the same Port ID value (e.g., a Port ID may support both iWARP and RoCE). If the Generate Exported Port ID bit in the Create operation is set to '1', then this field shall be ignored by the controller.
259:258	**Port ID of the Underlying Port** (refer to Figure 264)
291:260	**Transport Service ID (TRSVCID):** Specifies the NVMe Transport service identifier as an ASCII string. The NVMe Transport service identifier is specified by the associated NVMe Transport specification.

319:292	Reserved

Upon successful completion of a Manage Exported Port command with a Create operation, Dword 0 of the completion queue entry of the Manage Exported Port command contains the Exported Port ID associated with the Exported NVM Subsystem (refer to Figure TBD22).

Figure TBD22: Create Completion Queue Entry Dword 0 Data Structure

Bytes	Description
01:00	**Exported Port ID (EPID):** If the Generate Exported Port ID bit is set to '1' in the Create operation, then this field specifies Exported Port ID to be used for this Exported Transport and associated with the Exported NVM Subsystem identified by the Exported NVM Subsystem NVMe Qualified Name field in the Create data structure. If the Generate Exported Port ID bit is cleared to '0' in the Create operation, this field shall be set to the Exported Port ID provided in the Create data structure.
03:02	Reserved

If the Exported NVM Subsystem NVMe Qualified Name field specified in the Create Data Structure does not identify an existing Exported NVM Subsystem, then the controller shall abort the command with a status code of Invalid Field in Command.

If the Generate Exported Port ID bit is cleared to '0' in the Create data structure indicating an Exported Port ID is specified in the Manage Exported Port data structure and used to identify the new Exported Port; and:

- an Exported Port ID is not specified in the Manage Exported Port data structure in the data buffer; or
- an Exported Port ID specified in the data buffer does not uniquely identify an Exported NVM Subsystem Port (i.e., if the specified Exported Port ID is used to identify an already existing Exported NVM Subsystem Port),

then the controller shall abort the command with a status code of Invalid Field in Command.

5.TBD5.1.2 Delete (Management Operation 03h)

The Delete operation of the Manage Exported Port command is used to remove an Exported Port from an Exported NVM Subsystem and delete the Exported Port.

The Exported Port to be deleted is specified in the Exported Port ID (EPID) field in the Delete operation data structure (refer to Figure TBD23) in the data buffer.

The Exported NVM Subsystem NQN to which this Exported Port ID is assigned is specified in the NVM Subsystem NVMe Qualified Name (SUBNQN) field of the Delete operation data structure in the data buffer.

The Exported Port that is deleted by a Delete operation should not be in use (i.e., there should be no association between any host and a controller through that Exported Port). If the Exported Port is in use, then all associations between any host and a controller through that port are terminated by the Delete operation, and for any association that is terminated, the behavior of outstanding commands submitted via that association and resources underlying that association (e.g., fabric connections) is undefined.

Figure TBD23: Delete Data Structure

Bytes	Description
255:00	**NVM Subsystem NVMe Qualified Name (SUBNQN):** NVMe Qualified Name (NQN) that uniquely identifies the Exported NVM subsystem which is associated with the Exported Port to be deleted in command. Refer to section 4.5.
257:256	**Exported Port ID (EPID):** Specifies a particular NVM subsystem port to be deleted in this command.
319:258	Reserved

If:
- the Exported port ID field in the Delete operation data structure does not contain an Exported Port ID associated with the Exported NVM Subsystem NQN (SUBNQN) specified in the Delete data structure (refer to figure TBD23); or
- the NVM Subsystem NVMe Qualified Name field in the Delete data structure (refer to figure TBD23) does not contain an NVM Subsystem NQN for an existing Exported NVM Subsystem that is associated with the Exported Port ID specified in the Delete data structure,

then the controller shall abort the command with a status code of Invalid Field in Command.

Upon successful completion of a Manage Exported Port command with a Delete operation the specified Exported Port ID has been removed from the specified Exported NVM Subsystem.

5.TBD5.2 Command Completion

Upon completion of the Manage Exported Port command, the controller posts a completion queue entry to the Admin Completion queue indicating the status of the command. Refer to section 8.TBD1 for usages.

Modify section 8.3 to state that Capacity Management commands are prohibited in Exported NVM Subsystems.

8 Extended Capabilities

…

8.3 Capacity Management

Capacity Management is a capability for organizing physical media into Endurance Groups and NVM Sets. There are two different forms of Capacity Management, Fixed capacity Management and Variable Capacity Management. A controller that supports Capacity Management shall support at least one form.

Capacity Management commands shall not be supported by Exported NVM Subsystems.

…

Modify section 8.11 to slightly alter the title of the section, move text defining Namespace Management capabilities for NVM Namespaces that are not Exported NVM Namespaces from 8.11 to a new section numbered 8.11.TBD1 at the top of 8.11 (note – the text moved from 8.11 to 8.TBD1 has NOT changed – just the location shifted slightly) and add a new subsection numbered 8.11.TBD2 that defines Namespace Management capabilities for Exported NVM Subsystems.

8.11 Namespace Management

The Namespace Management capability consists of the Namespace Management command (refer to section 5.23) and the Namespace Attachment command (refer to section 5.22). The Namespace Management command is used to create a namespace or delete a namespace. The Namespace Attachment command is used to attach and detach controllers from a namespace. The Namespace Management capability is intended for use during manufacturing or by a system administrator.

If the Namespace Management capability is supported, then the controller:
   a) shall support the Namespace Management command and the Namespace Attachment command;

b) shall set bit 3 to '1' in the OACS field (refer to Figure 275);
c) should support the Namespace Attribute Changed asynchronous event (refer to Figure 146 and section 5.27.1.8); and
d) may support Namespace Granularity (refer to the NVM Command Set Specification).

A controller may support the Namespace Attachment command without supporting the Namespace Management command. Such a controller:
a) shall not set bit 3 to '1' in the OACS field (refer to Figure 275); and
b) should support the Namespace Attribute Change asynchronous event.

If a namespace is detached from a controller, then the NSID that referred to that namespace becomes an inactive NSID (refer to section 3.2.1.4) on that controller. If a namespace is deleted from the NVM subsystem, then the NSID that referred to that namespace becomes an unallocated NSID (refer to section 3.2.1.3) in the NVM subsystem. Previously submitted but uncompleted or subsequently submitted commands to the affected NSID are handled by the controller as if they were issued to an inactive NSID (refer to Figure 87).

The size of a namespace is based on the size 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.

Reporting of capacity information for the NVM subsystem, Domain, Endurance Group, and NVM Set are described in section 3.8. For each namespace, the NVM Set and the Endurance Group in which the namespace is allocated are 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.

If the NVM Set Identifier field and the Endurance Group Identifier field are both cleared to 0h in a create operation, then the controller shall choose the Endurance Group and the NVM Set from which to allocate the namespace.

If the NVM Set Identifier field is cleared to 0h and the Endurance Group Identifier field is set to a non-zero value in a create operation, then the controller shall choose the NVM Set in the specified Endurance Group from which to allocate the namespace.

If the NVM Set Identifier field is set to a non-zero value and the Endurance Group Identifier field is cleared to 0h in a create operation, then the controller shall abort the command with a status code of Invalid Field in Command.

If the NVM Set Identifier field and the Endurance Group Identifier field are both set to non-zero values in a create operation and the specified NVM Set exists in the specified Endurance Group, then the controller shall allocate the namespace in the specified NVM Set.

If the NVM Set Identifier field and the Endurance Group Identifier field are both set to non-zero values in a create operation and the specified NVM Set does not exist in the specified Endurance Group, then the controller shall abort the command with a status code of Invalid Field in Command.

For each namespace, the NVM capacity used for that namespace is reported in the Identify Namespace

data structure (refer to the applicable I/O Command Set specification). 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 the NVM Command Set Specification), if supported. For example, when using the NVM Command Set, 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 280 and the Identify Namespace data structure in the NVM Command Set Specification) is cleared to 0h, then the controller shall determine the ANAGRPID that is assigned to that namespace.

To create a namespace, if supported, host software performs the following actions:
1. Host software requests the Identify Namespace data structure that specifies common namespace capabilities (i.e., using an Identify command with the NSID field set to FFFFFFFFh and the CNS field cleared to 0h);
2. If the controller supports reporting of I/O Command Set specific Namespace Management content (refer to the Namespace Management section in the applicable I/O Command Set specification), host software optionally requests that information (e.g. Namespace Granularity).
3. Host software determines available capacity (refer to section 3.8);
4. Host software creates the data structure defined in Figure 300 (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 structures for the new namespace to determine all attributes of the namespace.

To attach a namespace, if supported, host software performs the following actions:
1. Host software issues the Namespace Attachment command specifying the Controller Attach operation to attach the specified namespace to one or more controllers; and
2. If Namespace Attribute Notices are enabled, the controller(s) newly attached to the namespace report a Namespace Attribute Changed asynchronous event to the host.

To detach a namespace, if supported, host software performs the following actions:
1. Host software issues the Namespace Attachment command specifying the Controller Detach operation to detach the specified namespace from one or more controllers; and
2. If Namespace Attribute Notices are enabled, the controllers that were detached from the namespace report a Namespace Attribute Changed asynchronous event to the host.

To delete a namespace, if supported, host software performs the following actions:
1. Host software should detach the namespace from all controllers;
2. Host software issues the Namespace Management command specifying the Delete operation for the specified namespace. On successful completion of the command, the namespace has been deleted; and
3. If Namespace Attribute Notices are enabled, any controller(s) not processing the Namespace Management command that was attached to the namespace reports a Namespace Attribute Changed asynchronous event to the host.

8.11.1 Namespace Management Considerations for Exported NVM Subsystems (informative)

A host is able to determine whether an Exported NVM Subsystem supports the Namespace Attachment command by reading the Commands Supported and Effects log page (refer to section 5.16.1.6). The Namespace Attachment command is used to attach namespaces to controllers and detach namespaces from controllers. The Namespace Management command is not supported by controllers in an Exported NVM subsystem.

If an Exported Namespace is detached from a controller in the Exported NVM Subsystem, then the NSID that referred to that namespace becomes an inactive NSID (refer to section 3.2.1.4) on that controller. If an Underlying Namespace is disassociated from the Exported NVM Subsystem (refer to section 5.TBD4.1.2), then the NSID of the Exported Namespace that referred to that namespace becomes an unallocated NSID (refer to section 3.2.1.3) and is not available to any controller in the Exported NVM Subsystem. Previously submitted but uncompleted or subsequently submitted commands to the namespace that is:
- detached from a controller; or
- disassociated from the Exported NVM Subsystem,

are handled by the controller as if they were issued to an inactive NSID (refer to section 3.2.1 and Figure 87). Refer to section 8.11 for Host actions to attach or detach a specified Namespace.

If an Exported NVM Subsystem Exports an Underlying Namespace that becomes unavailable (e.g., detached or deleted) or is affected by a Capacity Management command (e.g., Endurance Group deletion, NVM Set deletion (refer to section 8.3)) then:
- Previously submitted but uncompleted or subsequently submitted commands to the Exported Namespace are handled by the controller as if they were issued to an inactive NSID (refer to section 3.2.1 and Figure 87); and
- If Namespace Attribute Notices are enabled controllers affected by Underlying Namespace changes report a Namespace Attribute Changed asynchronous event to the host as described in Figure 146.

Add new section 8.21.2 to specify the effects of Sanitize operations on Exported NVM Subsystems

8.21   Sanitize Operations

…

8.21.TBD1 Sanitize Operation Effects on Exported NVM Subsystems

Performing a sanitize operation on an Underlying NVM Subsystem (refer to section 8.TBD1) sanitizes user data in all Underlying Namespaces contained in that NVM subsystem, including any Underlying Namespace that is associated with an Exported Namespace in any Exported NVM Subsystem (refer to section 5.TBD4).

If an Exported NVM Subsystem contains an Exported Namespace that is associated with an Underlying Namespace in an Underlying NVM Subsystem for which one of the following conditions exists:
- a sanitize operation is in progress; or
- the most recent sanitize operation has failed and successful recovery from the failed sanitize operation has not occurred,

then, while that condition exists, that Exported NVM Subsystem shall enforce the I/O command sanitize operation restrictions described in section 8.21.1 on I/O commands that specify that Exported Namespace and may enforce additional sanitize operation restrictions described in that section.

Add new sections 8.TBD1 – 8.TBD3 to describe exporting NVM resources and show example flows using the added commands and associated details.

8.TBD1 Exporting NVM Resources

Exported Resource Management is a capability that may be supported in an NVM subsystem that only contains controllers that use transports other than the Memory-Based Transport Model (e.g., the PCIe transport).

Exporting an Underlying Namespace may be achieved by:
- creating an Exported NVM Subsystem (refer to section 5.TBD2);
- assigning an Underlying Namespace to an Exported NVM Subsystem (refer to section 5.TBD4.1.1);
- attaching a transport to the Exported NVM Subsystem to enable remote access to the Exported NVM Subsystem (refer to section 5.TBD5.1.1); and
- optionally assigning access control policies for the Exported NVM Subsystem (refer to sections 5.TBD3.1.3 and 5.TBD3.1.4).

Exported NVM Subsystems shall not expose information about Underlying NVM Subsystem resources or entities that are not associated with Exported NVM Subsystem entities. This includes entities in Underlying NVM Subsystems (e.g., NSID, NVM Set Identifiers, ANA Group Identifiers).

8.TBD1.1 Management of Exported NVM Resources (Informative)

Commands for managing Exported NVM Resources are processed by a controller in an Underlying NVM Subsystem.

This section describes example flows to manage Exported NVM Resources through the use of the:
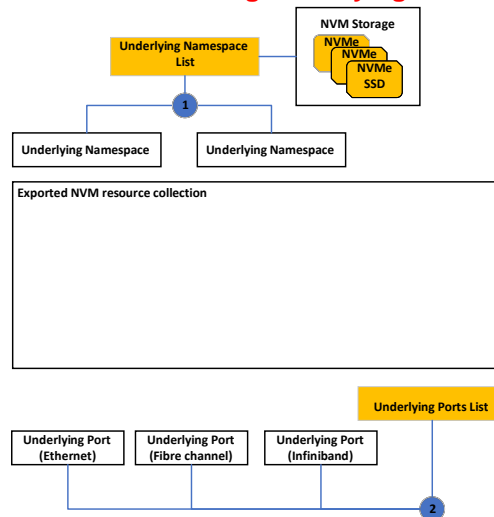- Identify command with CNS set to 1Dh to retrieve the Underlying Namespace List;
- Identify command with CNS set to 1Eh to retrieve the list of Underlying Ports that may be used to export NVMe over Fabrics Subsystems;
- Create Exported NVM Subsystem command to create an Exported NVM Subsystem;
- Manage Exported NVM Subsystem command to manage access to an Exported NVM Subsystem;
- Manage Exported Namespace command to configure an Exported NVM Subsystem with namespaces;
- Manage Exported Port command to manage fabric port configurations on an Exported NVM Subsystem;
- Manage Exported NVM Subsystem command to delete Exported NVM Subsystems; and
- Clear Exported NVM Resource Configuration command to remove all configured Exported NVM Resources.

8.TBD1.1.1 Configuring an Exported NVM Subsystem

Prior to configuring exported NVM resources and exposing them for access, an administrative entity (e.g., administrator, resource manager, orchestrator, administration console, centralized configuration manager) must first determine what Underlying resources (e.g., namespaces, ports) are available; this may be

through a-priori knowledge or by using a Identify command with CNS set to 1Dh (refer to section 5.17.2.TBD1 (refer to step '1' in Figure TBD24, Figure TBD25, and in Figure TBD26) and Identify command with CNS set to 1Eh (refer to section 5.17.2.TBD2) (refer to step '2' in Figure TBD24, Figure TBD25, and Figure TBD26). Figure TBD24 shows collections of Underlying Namespaces and Underlying Ports and an empty collection of Exported NVM resources.

Figure TBD24: Example reference for retrieving Underlying Namespaces and Underlying Ports

Once the administrative entity has determined the available underlying resources (refer to step '1' and step '2' in Figure TBD24, Figure TBD25, and Figure TBD26), the administrative entity may create, configure, and expose Exported NVM Subsystems. An example flow to configure an Exported NVM Subsystem:

- Create an Exported NVM Subsystem by Issuing a Create Exported NVM Subsystem command (refer to section 5.TBD2) (refer to step '3' in Figure TBD25 and Figure TBD26).
- Associate individual Underlying Namespaces with an Exported NVM Subsystem by issuing a Manage Exported Namespace command with an Associate Namespace operation (refer to section 5.TBD4.1.1) (refer to step '4' in Figure TBD25 and Figure TBD26).
- Associate individual Underlying Ports with an Exported NVM Subsystem by issuing a Manage Exported Port command with a Create operation (refer to section 5.TBD5.1.1) (refer to step '5' in Figure TBD25 and Figure TBD26).

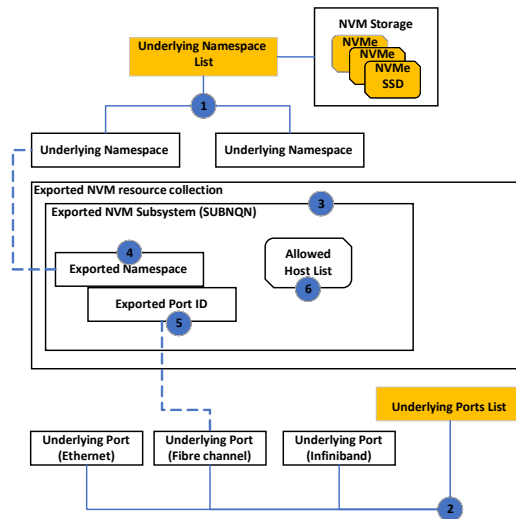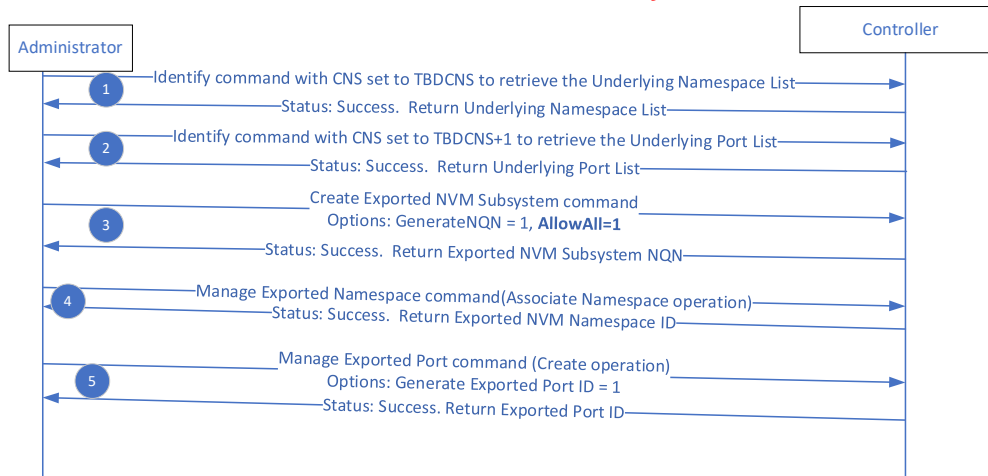**Figure TBD25: Example reference for creating and configuring an Exported NVM Subsystem**

Figure <mark>TBD26</mark>: Example steps for retrieving underlying resources and configuring an Exported NVM Subsystem
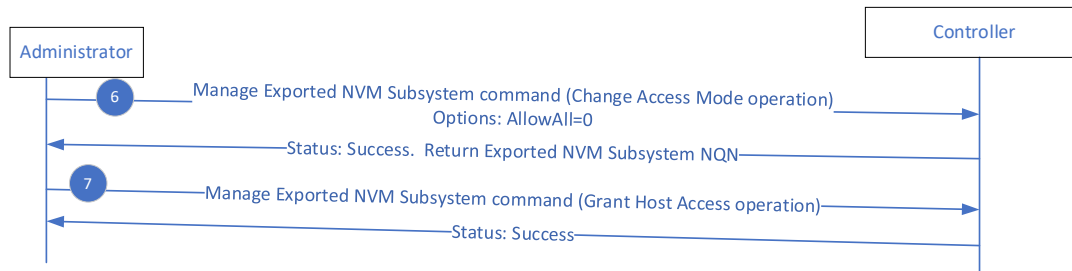

8.TBD1.1.2 Managing Host access to an Exported NVM Subsystem

Exported NVM Subsystems are created with a specified Access Mode (i.e., restricted, unrestricted).

This section describes an example flow for changing an Exported NVM Subsystem configured for unrestricted access to restricted access (where only hosts in the Allowed Host List associated to the specified Exported NVM Subsystem have permissions to use the Exported Namespaces linked to that Exported NVM Subsystem).
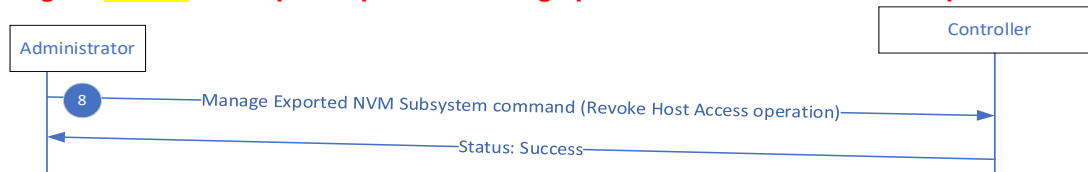
The access mode of Exported NVM Subsystems may be changed by issuing a Manage Exported NVM Subsystem command with a Change Access Mode operation (refer to section <mark>5.TBD3.1.2</mark>) and specifying the desired Access Mode (refer to step '6' in Figure <mark>TBD27</mark>). The example Figure <mark>TBD27</mark> shows changing the access mode of an Exported NVM Subsystem to restricted access. The example Figure <mark>TBD27</mark> also shows Exported NVM Subsystem access to specific hosts (refer to step '7' in Figure <mark>TBD27</mark>).

Figure <mark>TBD27</mark>: Example steps for restricting access to an Exported NVM Subsystem to specific hosts

Similarly, hosts may be restricted from access to specified Exported NVM Subsystems configured for restricted access, on specified ports, by issuing Manage Exported NVM Subsystem commands with the Revoke Host Access operation (refer to section 5.TBD3.1.4). Refer to step '8' in Figure TBD28).
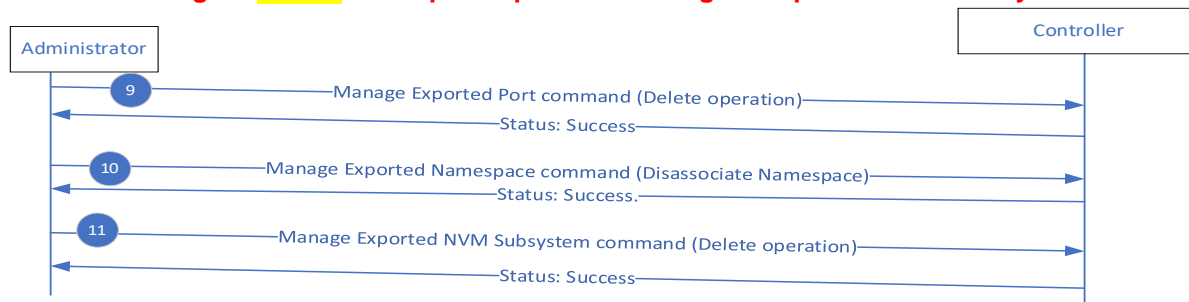
Figure TBD28: Example steps for revoking specific hosts access to an Exported NVM Subsystem

An example flow for removing an Exported NVM Subsystem and associated Namespaces would be issuing:

1. a Manage Exported NVM Port command with a Delete operation (refer to section 5.TBD5.1.2) for each port associated with the Exported NVM Subsystem to be removed (refer to step '9' of Figure TBD29).
2. a Manage Exported Namespace command with a Disassociate Namespace operation (refer to section 5.TBD4.1.2) for each Exported Namespaces associated with the Exported NVM Subsystem to be removed (refer to step '10' of Figure TBD29).
3. a Manage Exported NVM Subsystem command with a Delete operation (refer to section 5.TBD3.1.1) of a Manage Exported NVM Subsystem command (refer to step '11' of Figure TBD29).

Figure TBD29: Example steps for removing an Exported NVM Subsystem

8.TBD1.1.3 Clearing all Exported NVM Subsystems

Issuing a Clear Exported NVM Resource Configuration command (refer to section 5.TBD1) clears all Exported NVM Subsystems (refer to Figure TBD30).

Figure TBD30: Example steps for clearing all Exported NVM Subsystems

Administrator

Controller

Clear Exported NVM Resource Configuration command

Status Success

Upon successful completion of a Clear Exported NVM Resource Configuration command there are no Exported NVM Subsystems.

Description for Changes Document for NVMe-MI Specification 1.2

Defines new administrative commands
References NVMe-MI Specification 1.2 section 6

Description of Specification Changes

Modify section 6 to add the new commands to Figure 114: List of NVMe Admin Commands Supported using the Out-of-Band Mechanism.

6 NVM Express Admin Command Set

The NVM Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. Figure 114 shows NVMe Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device and an NVMe Enclosure using the out-of-band mechanism. All NVMe Admin Commands are prohibited using the in-band tunneling mechanism. The commands are defined in the NVM Express Base Specification and the I/O Command Set specifications. If an NVMe Admin Command is issued in a Request Message that is a prohibited command in Figure 114, the Management Endpoint shall return an Invalid Parameter Error Response with PEL field indicating the NVMe opcode. Future revisions of this specification may add additional commands to Figure 114. The NVM Express Admin Command Set is only applicable in the out-of-band mechanism and is prohibited in the in-band tunneling mechanism.

Figure 114: List of NVMe Admin Commands Supported using the Out-of-Band Mechanism

Command	Opcode	NVMe Storage Device O/M/P[1]	NVMe Enclosure O/M/P[1]	Reference Specification
Abort	00h	P	P	NVMe Base Specification
Asynchronous Event Request	0Ch	P	P	NVMe Base Specification
Capacity Management	20h	O	P	NVMe Base Specification
…	…	…	…	…
Clear Exported NVM Resource Configuration	26h	O	O	NVM Express Base Specification
Create Exported NVM Subsystem	2Ah	O	O	NVM Express Base Specification
Manage Exported NVM Subsystem	2Dh	O	O	NVM Express Base Specification
Manage Exported Namespace	31h	O	O	NVM Express Base Specification
Manage Exported Port	35h	O	O	NVM Express Base Specification
…	…	…	…	…
Virtualization Management	1Ch	O	O	NVMe Base Specification
Fabrics Commands	7Fh	P	P	NVMe Base Specification

NOTES:
1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited from being supported. An NVMe Enclosure that is also an NVMe Storage Device (i.e., implements Namespaces) shall implement mandatory commands required by either an NVMe Storage Device or an NVMe Enclosure and may implement optional commands allowed by either an NVMe Storage Device or an NVMe Enclosure. Mandatory commands shall be supported using the out-of-band mechanism if the NVMe Controller specified by the Controller ID field supports the command inband.

2. If the Retain Asynchronous Event bit is cleared to '0', then the status associated with the NVMe Admin Command shall be Invalid Field in Command (i.e., the NVMe Admin command is aborted). For implementations compliant to version 1.1 or earlier of this specification, the Retain Asynchronous Event bit in the Get Log Page command (refer to the NVM Express Base Specification) may or may not be ignored by the Controller. Refer to section 6.2