



LEGAL NOTICE:

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

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

LEGAL DISCLAIMER:

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

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

The NVM Express® design mark is a registered trademark of NVM Express, Inc.

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

NVM Express Technical Proposal for New Feature

Technical Proposal ID	TP4034
Change Date	2021-09-14
Builds on Specification	NVM Express Base 1.4c NVMe over Fabrics 1.1a NVM Express Management Interface 1.2 NVM Express Base 2.0 NVM Command Set 1.0

Technical Proposal Author(s)

Name	Company
David Black	Dell EMC
Curtis Ballard, Matt Goepfert	HPE
Fred Knight	NetApp

This technical proposal defines new requirements and capabilities that enable two or more NVM subsystems to present the same namespace. When the same namespace is presented by two or more NVM subsystems, this TP proposes that the namespace be referred to as a dispersed namespace. Online data migration and data replication are two such scenarios that would utilize dispersed namespaces.

Failure of hosts to properly identify a namespace dispersed across multiple NVM subsystems as the same namespace can result in data corruption. This TP seeks to prevent data corruption due to host failure to properly recognize dispersed namespaces, as well as controller failure to properly support access to dispersed namespaces.

<This TP includes changes for NVMe Base 1.4c and NVMe-oF 1.1a as well as technically identical changes for the NVMe 2.0 Library (i.e., NVMe-MI 1.2, NVMe Base 2.0, and NVM Command Set 1.0).>

Revision History

Revision Date	Change Description
2020-04-06	Initial version
2020-05-21	Incorporated Dell EMC comments/changes
2020-06-09	Accepted changes/removed comments for non-open discussion items
2020-06-24	Incorporated multiple changes based upon FMDS discussions
2020-07-21	Incorporated multiple changes based upon FMDS discussions (primarily ANA-related material)
2020-08-18	Closed on open ANA consideration discussions, removed Host ID-Host NQN mapping requirement for controllers (and the associated new error code) and replaced it with unique 128-bit Host ID requirement for hosts, and added prohibition of 64-bit Host IDs

2020-08-25	Closed on requiring reservation key registrations and reservation key value replacements to be coordinated across participating NVM subsystems and cleaned up the wording for both in 8.8.2. Removed the word “coherent” from 8.NEW. Closed on Host Dispersed Namespace Support Not Enabled status code being a Generic Command status code
2020-09-09	Closed on prohibiting all I/O commands and determined which specific Admin commands also need to be prohibited (table added to section 5.NEW.3) while HDISNS cleared to 0h
2020-11-03	<ul style="list-style-type: none"> • Incorporated various phase 3 edits from Mike Allison • Added names for bits • Removed Namespace Management command restrictions for deleting dispersed namespaces (and the associated error code) • Removed 128-bit Host ID requirement • Added clarification for “shared set” of dispersed namespaces for ANA groups
2020-11-12	<ul style="list-style-type: none"> • Removed MNSUB + ANSUB fields and all related text • Changed new log page’s name, removed its relation to ANSUB field, removed its indexing, and added a header to it consisting of GENCTR + NUMPSUB fields • Made FFFDh the CNTLID used in Registered Controller and Registered Controller Extended data structures when the controller is contained in a separate participating NVM subsystem • Created a new table in NVMe-oF 1.1 section 4.2 to describe CNTLIDs FFF0h to FFFFh and added a reference to it in NVMe-oF 1.1 section 3.3 • Added a bullet to Connect Invalid Parameters in NVMe-oF 1.1 section 3.3 prohibiting CNTLIDs FFF0h to FFFFh from being specified in the Connect command • Removed “special value” from existing text in NVMe-oF 1.1 section 4.2 (two occurrences) • Various wording changes
2020-12-03	<ul style="list-style-type: none"> • Reworded second paragraph in section 8.NEW.6 to add clarity
2021-01-20	<ul style="list-style-type: none"> • Added optional support for old hosts (i.e., HDISNS = 0h) to access dispersed namespaces from a single participating NVM subsystem
2021-02-02	<ul style="list-style-type: none"> • Cleaned up wording in Figure 315, section 8.NEW.3.1, and Figure NEW.5 • Set Features command now prohibited (not just 3 specific FIDs) when specifying NSID of dispersed namespace when HDISNS != 1h
2021-02-03	<ul style="list-style-type: none"> • Fixed a mistake in modification to NVMe-oF 1.1 section 3.3: a host specifying a Controller ID of FFFEh or FFFFh in a Connect command does not result in Connect Invalid Parameters being returned (only FFF0h – FFFDh do) • Fixed a typo and added clarification to Figure NEW.1 in NVMe-oF 1.1 section 4.2
2021-02-11	<ul style="list-style-type: none"> • Added new bit (DISNSRS) to all reservation commands, which dispersed namespace supporting hosts (i.e., HDISNS = 1h) can set to ‘1’ to signal support for reservations on dispersed namespaces (i.e., it’s OK to receive CNTLID = FFFDh for controllers in “separate” participating NVM subsystems) • Added text to section 8.NEW.6 that references new bit and adds new related controller requirement to abort reservation commands where HDISNS = 1h and DISNSRS != ‘1’
2021-02-16	<ul style="list-style-type: none"> • Fixed wording in DISNSRS bit and section 8.NEW.6 • Created new command specific status code (Namespace Is Dispersed) applicable to all reservation commands • Added DISNSRS validation requirement to list of controller behavior related to HDISNS in section 8.NEW.3

2021-02-23	<ul style="list-style-type: none"> • Cleaned up section 8.NEW.2 to add clarity, turn a few “mays” into “shalls”, and give precedence to NGUID over UUID when both are present • Added a new controller requirement into section 8.NEW.3 that when HDISNS = 1h the controller shall support the new DISNSRS bit in the reservation commands being set to ‘1’ • Cleaned up “old host” support paragraph in section 8.NEW.3 and added precision to the use of “Host NQN” throughout this section • Added comments from Mike Allison
2021-03-02	<ul style="list-style-type: none"> • Moved DISNSRS bit support controller requirement out of HDISNS-related list in section 8.NEW.3 and into section 8.NEW.6 • Added reference in section 8.8 (Reservations) for DISNSRS bit support controller requirement model clause text
2021-03-08	<ul style="list-style-type: none"> • Added changes from David Black related to collapsing multiple identical records for separate controllers (i.e., CNTLID = FFFDh) into a single Registered Controller data structure. See section 6.13, Figure 394, and section 8.NEW.6 • Deleted modifications to Figure 264, thus allowing hosts to specify that a namespace has the capability of being a dispersed namespace upon creation. The actual means of making that namespace dispersed (e.g., how it becomes contained on other subsystems) is out of scope • Changed wording in 8.NEW.3 to align with wording in new bit in NMIC – i.e., DISNS = ‘1’ means the namespace has the capability of being dispersed, not that it is in fact a dispersed namespace contained on multiple participating subsystems • Changed “may” to “shall either” in section 8.NEW.3 for how a second participating subsystem reacts to a Connect command from an “old host” • Changed “would” to “will” in text in section 8.NEW.3 dealing with “old host” connectivity to a second participating subsystem
2021-03-15	<ul style="list-style-type: none"> • Fixed standard-ese in section 8.NEW.6 for new text pertaining to collapsing multiple identical FFFDh records • Fixed wording in 8.NEW.1 and 8.NEW.3 to make the distinction between a namespace that is capable of being a dispersed namespace and the actual act of dispersing that namespace
2021-03-17	<ul style="list-style-type: none"> • Added numbers to the ANA Groups in Figure NEW.2 to make it more clear that they are different ANA Groups across both NVM subsystems • Added Dispersed Namespace Participating NVM Subsystems log page to table TP 6022 added to NVMe-MI 1.2 Member Review Draft • Doc is now based off NVMe 1.4b instead of 1.4a
2021-03-18	<ul style="list-style-type: none"> • Clarified that NVMe-MI 1.2 is not a released spec. TP 4034 will eventually align with NVMe 2.0 before it reaches 30-day member review
2021-03-26	<ul style="list-style-type: none"> • Made minor edits to text throughout the doc to move the value of a bit/field to be directly adjacent to the bit/field itself • Added an informational sentence to section 8.NEW.3 to warn readers that choosing the optional subsystem behavior of blocking an old host’s Connect command to a second participating subsystem can impact host access to unrelated namespaces on that second participating subsystem • Split former Figure NEW.4 into two separate data replication figures • Added a third data replication figure related to failover functionality • Modified Figure NEW.2 to provide a more complex example and illustrate partially overlapping dispersed namespaces

2021-04-05	<ul style="list-style-type: none"> Modified Figure NEW.2 to eliminate unnecessary complexity of depicting ANA Groups that don't contain dispersed namespaces Modified text in section 8.20.2 to accommodate changes made to Figure NEW.2 Tied the NQNs from the new log page with the NQNs from the SUBNQN field in Identify Controller data structures
2021-04-08	<ul style="list-style-type: none"> Removed specifying NSID = FFFFFFFFh as an option for the new log page (no valid use cases can be identified for it) Added that asking for the new log page for a non-dispersed namespace will result in Invalid Field in Command Modified dispersed namespace definition to match wording used for shared namespace definition Modified text in DISNS bit in NMIC field to not duplicate dispersed namespace definition Added text in section 8.NEW.3 explaining the uses for option a vs. option b for how subsystems handle old hosts
2021-04-14	<ul style="list-style-type: none"> Modified headings throughout the doc to match the predominant style used in TPs Moved text from section 8.NEW.4 into section 8.NEW.3 Marked a few reserved values that were previously overlooked as TBD for places where new values are being inserted
2021-04-21	<ul style="list-style-type: none"> Added "Unique Identifier Overview" section (section 7.11.1) and bumped each of the following sections in 7.11 up one number Restructured Format NVM text in Figure NEW.6
2021-05-19	<ul style="list-style-type: none"> Fixed typo in DISNSRS field in figures 382, 386, 389, and 393
2021-05-25	<ul style="list-style-type: none"> Added clarification to CNTLID FFFDh text in section 8.NEW.6
2021-06-08	<ul style="list-style-type: none"> Refactoring changes other than simple figure/section number changes: <ul style="list-style-type: none"> DISNS bit in NMIC field of Identify Namespace data structure can be obtained from either CNS 00h (if supported by the I/O command set) or CNS 08h I/O commands reference includes both those in section 7 of the base spec and those in I/O commands section in each I/O Command Set spec NVM Command Set Spec now included in this TP because modified atomic operations text now lives in that spec due to refactoring
2021-06-17	<ul style="list-style-type: none"> Changed "this field" to "this bit" for DISNSRS bit Slight modification to Identify command mentioned in section 8.NEW.3 Slight modification to I/O commands reference
2021-06-30	<ul style="list-style-type: none"> Removed requirement that HDISNS = 1 before controller can return CNTLID = FFFDh for reservations held on controllers in separate participating NVM subsystems (section 8.NEW.3)
2021-07-13	<ul style="list-style-type: none"> Removed ANA grouping requirements for overlapping set of dispersed namespaces (section 8.1.2)
2021-07-22	<ul style="list-style-type: none"> Accepted tracked changes after TWG review – ready for 30-day member review
2021-08-24	<ul style="list-style-type: none"> Changed document name to reflect pre-integration status Changed multiplication sign from "x" to "*" in Figure NEW.1 (twice) Removed an extra space in Figure 351 Changed "can be" to "is able to be" in section 8.19.3 and 8.NEW.5 Changed "can be" to "is allowed to be" in Figure NEW.A (twice)
2021-08-25	<ul style="list-style-type: none"> Changed NVMe Base 1.4b version to be based on NVMe Base 1.4c Changed NVMe-oF 1.1 version to be based on NVMe-oF 1.1a Added note that pre-2.0 versions of specs are included for ease of implementation

2021-09-01	<ul style="list-style-type: none"> Integrated into the NVM Express Management Interface Specification, revision 1.2, the NVM Express Base Specification, revision 2.0, and the NVM Command Set Specification, revision 1.0.
2021-09-09	<ul style="list-style-type: none"> Filled in the TBDs for the NVMe 1.4c section Added NVM Base 1.4c and NVMe-oF 1.1a specs to “Builds on Specification” section
2021-09-10	<ul style="list-style-type: none"> Added change list
2021-09-14	<ul style="list-style-type: none"> Changed “Specific” to “specific” in section 8.8.3. Removed highlight from text that was filled in Clarified figure 317 for NVMe 1.4c changes.

Description for NVMe-MI 1.2, NVMe Base 2.0, and NVM Command Set 1.0 Changes Document

NVMe-MI 1.2:

- Added the Dispersed Namespace Participating NVM Subsystems log page to the Management Endpoint – Log Page Support table as optional
- New requirement / incompatible change:**
 - None
- References:
 - Section 6.3
 - TP 4034

NVMe Base 2.0:

- Added new definitions for dispersed namespaces and participating NVM subsystems
- Added a new optional log page called Dispersed Namespace Participating NVM Subsystems
 - Added it to the log page support tables for all three controller types, as optional for I/O and Admin controllers and prohibited for Discovery controllers
- Allocated Controller ID FFFDh to be used in the CNTLID field of Registered Controller data structures when the controller associated with the reservation is contained in a separate participating NVM subsystem
 - Added a new table to define the use of Controller ID values FFF0h to FFFFh
- Added a new Host Behavior Support feature field called Host Dispersed Namespace Support (HDISNS) that a host sets to signal that it supports accessing dispersed namespaces
 - Added a generic status code called Host Dispersed Namespace Support Not Enabled to be used when a host that has not set HDISNS to ‘1’ sends an I/O command or a select Admin command (see Figure NEW.7) to a dispersed namespace
- Added a new bit to each reservation command called Dispersed Namespace Reservation Support (DISNSRS) that a host sets to signal that it supports using reservations on dispersed namespaces
 - Added a new command specific status code called Namespace Is Dispersed to be used when a host that has set HDISNS to ‘1’ sends a reservation command to a dispersed namespace without setting the DISNSRS bit to ‘1’ in that reservation command
- Broke section 4.5.1 out into separate sub-sections for each different type of unique identifier
- Added a new bit to the I/O Command Set Independent Identify Namespace data structure in the NMIC field called Dispersed Namespace (DISNS) that is used to indicate that the namespace is a dispersed namespace
- Added a model clause section for dispersed namespaces (see section 8.NEW)
- New requirement / incompatible change:**
 - If a dispersed namespace is a member of an ANA Group on one participating NVM subsystem then it must be included in an ANA Group on all participating NVM subsystems
 - Hosts must use globally unique namespace identifiers when accessing dispersed namespaces upon setting HDISNS to ‘1’

- Hosts must use Host Identifiers that are unique across all hosts on all participating NVM subsystems upon setting HDISNS to '1'
- Hosts must set DISNSRS to '1' in each reservation command sent to a dispersed namespace upon setting HDISNS to '1'
- Controllers that support dispersed namespaces must support the DISNSRS bit being set to '1' in reservation commands
- If a reservation key is replaced on one participating NVM subsystem then it must be replaced on all participating NVM subsystems
- References:
 - Section 1.5
 - Section 3.1.2
 - Section 3.3.3
 - Section 4.5.1
 - Section 5.16.1
 - Section 5.17.2
 - Section 5.27.1
 - Section 6.3
 - Section 7.2
 - Section 7.3
 - Section 7.4
 - Section 7.5
 - Section 8.1.2
 - Section 8.10.1
 - Section 8.19
 - Section 8.NEW
 - TP 4034

NVM Command Set 1.0:

- Added a clarification that dispersed namespaces follow the same atomicity requirements as shared namespaces that are not dispersed namespaces
- **New requirement / incompatible change:**
 - None
- References:
 - Section 2.1.4
 - TP 4034

Markup conventions

Style	Meaning
Black	Unchanged existing material
Blue	New material to be added
Red	Existing material to be deleted
Blue	New material to be renumbered as incorporated
<Green>	Note to the editor

Description of NVMe 1.4c Base Specification Changes

1 Introduction

...

1.6 Definitions

...

1.6.NEW1 dispersed namespace

A shared namespace that may be concurrently accessed by controllers in two or more NVM subsystems (refer to section 8.NEW).

1.6.NEW2 participating NVM subsystem

An NVM subsystem that participates in (i.e., contains controllers that provide access to) a dispersed namespace.

4 Data Structures

...

4.6.1.2.1 Generic Command Status Definition

...

Figure 128: Status Code – Generic Command Status Values

Value	Description
...	...
22h	Transient Transport Error: A transient transport error was detected. If the command is retried on the same controller, the command is likely to succeed. A command that fails with a transient transport error four or more times should be treated as a persistent transport error that is not likely to succeed if retried on the same controller.
26h	Host Dispersed Namespace Support Not Enabled: The command is prohibited while the Host Dispersed Namespace Support (HDISNS) field is not set to 1h in the Host Behavior Support feature (refer to Figure 317).
23h 27h to 7Fh	Reserved
80h to BFh	I/O Command Set Specific
...	...

...

4.6.1.2.2 Command Specific Status Definition

...

Figure 130: Status Code – Command Specific Status Values

Value	Description	Commands Affected
...
25h	ANA Attach Failed	Namespace Attachment
...
2Eh	Namespace Is Dispersed	Reservation Acquire, Reservation Register, Reservation Release, Reservation Report
26h 2Fh to 6Fh	Reserved	
70h to 7Fh	Directive Specific	NOTE 1
...

...

5 Admin Command Set

...

5.14.1 Log Specific Information

...

Figure 195: Get Log Page – Log Page Identifiers

Log Identifier	Scope	Log Page Name	Reference Section
...
0Fh	NVM Subsystem	Endurance Group Event Aggregate	5.14.1.15
17h	Namespace	Dispersed Namespace Participating NVM Subsystems	5.14.1.NEW
40h 18h to 6Fh	Reserved		
70h	Discovery (refer to NVMe over Fabrics specification)		
...	...		
<p>KEY:</p> <p>Namespace = The log page contains information about a specific namespace.</p> <p>Controller = The log page contains information about the controller that is processing the command.</p> <p>NVM subsystem = The log page contains information about the NVM subsystem.</p>			
<p>NOTES:</p> <ol style="list-style-type: none"> For namespace identifiers of 0h or FFFFFFFFh. For namespace identifiers other than 0h or FFFFFFFFh. Bit 0 is cleared to '0' in the DSTO field in the Identify Controller data structure (refer to Figure 251). Bit 0 is set to '1' in the DSTO field in the Identify Controller data structure. Selection of a UUID may be supported. Refer to section 8.24. 			

...

5.14.1.NEW Dispersed Namespace Participating NVM Subsystems (Log Identifier 17h)

This log page is used to provide a list of NQNs for all participating NVM subsystems which contain controllers that are able to provide access to the dispersed namespace for the specified NSID. If an NSID of 0h or FFFFFFFFh is specified by the host in the Get Log Page command, then the controller shall abort that command with a status code of Invalid Namespace or Format as described in Figure 106. If the NSID specified by the host in the Get Log Page command is associated with a namespace that is not a dispersed namespace (i.e., the Dispersed Namespace (DISNS) bit is cleared to '0' in the Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC) field of the Identify Namespace data structure associated with that namespace), then the controller shall abort that command with a status code of Invalid Field in Command as described in section 5.14. The method used by the NVM subsystem containing the controller processing the command to discover the NQNs of separate participating NVM subsystems which contain controllers that are able to provide access to the same dispersed namespace is outside the scope of this specification. Each entry in the list shall contain an NQN that matches the NQN contained in the NVM Subsystem NVMe Qualified Name (SUBNQN) field of that participating NVM subsystem's Identify Controller data structure (refer to Figure 251). The first entry in the list shall always contain the NQN of the NVM subsystem containing the controller processing the command.

The log page returned is defined in Figure **NEW.1**.

Figure NEW.1: Get Log Page – Dispersed Namespace Participating NVM Subsystems Log

Bytes	Description
07:00	Generation Counter (GENCTR): This field contains a value that is incremented each time the participating NVM subsystem information for the specified NSID changes (i.e., each time participating NVM subsystems are added or removed for the specified NSID since the last time the host read this log page. 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 of Participating NVM Subsystems (NUMPSUB): This field indicates the number of participating NVM subsystems contained in the log.
255:16	Reserved
511:256	Participating NVM Subsystem Entry 0 (PSUBE0): This field contains the NQN of the NVM subsystem containing the controller processing the command.
767:512	Participating NVM Subsystem Entry 1 (PSUBE1): This field contains the NQN of a participating NVM subsystem that contains controllers that are able to provide access to the dispersed namespace.
...	...
$((N + 2) * 256) - 1$: $(N + 1) * 256$	Participating NVM Subsystem Entry N (PSUBEN): This field contains the NQN of a participating NVM subsystem that contains controllers that are able to provide access to the dispersed namespace.

...

5.15.2.1 Identify Namespace data structure (CNS 00h)

...

Figure 249: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M ¹	Description
...

Bytes	O/M ¹	Description								
29	M	<p>End-to-end Data Protection Type Settings (DPS): This field indicates the Type settings for the end-to-end data protection feature. Multiple bits may be set in this field. Refer to section 8.3.</p> <p>Bits 7:5 are reserved.</p> <p>Bit 4 if set to '1' indicates that the namespace supports protection information transferred as the last eight bytes of metadata. Bit 4 if cleared to '0' indicates that the namespace does not support protection information transferred as the last eight bytes of metadata.</p> <p>Bit 3 if set to '1' indicates that the namespace supports protection information transferred as the first eight bytes of metadata. Bit 3 if cleared to '0' indicates that the namespace does not support protection information transferred as the first eight bytes of metadata.</p> <p>Bit 2 if set to '1' indicates that the namespace supports Protection Information Type 3. Bit 2 if cleared to '0' indicates that the namespace does not support Protection Information Type 3.</p> <p>Bit 1 if set to '1' indicates that the namespace supports Protection Information Type 2. Bit 1 if cleared to '0' indicates that the namespace does not support Protection Information Type 2.</p> <p>Bit 0 if set to '1' indicates that the namespace supports Protection Information Type 1. Bit 0 if cleared to '0' indicates that the namespace does not support Protection Information Type 1.</p>								
30	O	<p>Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC): This field specifies multi-path I/O and namespace sharing capabilities of the namespace.</p> <table border="1" data-bbox="516 1056 1419 1623"> <thead> <tr> <th data-bbox="516 1056 834 1100">Bits</th> <th data-bbox="834 1056 1419 1100">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="516 1100 834 1144">Bits 7:4</td> <td data-bbox="834 1100 1419 1144">areReserved</td> </tr> <tr> <td data-bbox="516 1144 834 1356">1</td> <td data-bbox="834 1144 1419 1356">Dispersed Namespace (DISNS): If set to '1', then the namespace is a dispersed namespace (refer to section 8. NEW). If this bit is set to '1', then the namespace shall be a shared namespace (i.e., the Shared Namespace (SHRNS) bit shall also be set to '1'). If cleared to '0', then the namespace is not a dispersed namespace.</td> </tr> <tr> <td data-bbox="516 1356 834 1623">Bit0</td> <td data-bbox="834 1356 1419 1623">Shared Namespace (SHRNS): If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). If cleared to '0', then the namespace is a private namespace and is able to be attached to only one controller at a time. <Find all references to Bit 0 in the NMIC field throughout the spec and change them to "the Shared Namespace (SHRNS) bit".></td> </tr> </tbody> </table>	Bits	Description	Bits 7:4	are Reserved	1	Dispersed Namespace (DISNS): If set to '1', then the namespace is a dispersed namespace (refer to section 8. NEW). If this bit is set to '1', then the namespace shall be a shared namespace (i.e., the Shared Namespace (SHRNS) bit shall also be set to '1'). If cleared to '0', then the namespace is not a dispersed namespace.	Bit 0	Shared Namespace (SHRNS): If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). If cleared to '0', then the namespace is a private namespace and is able to be attached to only one controller at a time. <Find all references to Bit 0 in the NMIC field throughout the spec and change them to "the Shared Namespace (SHRNS) bit".>
Bits	Description									
Bits 7:4	are Reserved									
1	Dispersed Namespace (DISNS): If set to '1', then the namespace is a dispersed namespace (refer to section 8. NEW). If this bit is set to '1', then the namespace shall be a shared namespace (i.e., the Shared Namespace (SHRNS) bit shall also be set to '1'). If cleared to '0', then the namespace is not a dispersed namespace.									
Bit 0	Shared Namespace (SHRNS): If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). If cleared to '0', then the namespace is a private namespace and is able to be attached to only one controller at a time. <Find all references to Bit 0 in the NMIC field throughout the spec and change them to "the Shared Namespace (SHRNS) bit".>									

Bytes	O/M ¹	Description
31	O	<p>Reservation Capabilities (RESCAP): This field indicates the reservation capabilities of the namespace. A value of 0h in this field indicates that reservations are not supported by this namespace. Refer to section 8.8 for more details.</p> <p>Bit 7 if set to '1' indicates that Ignore Existing Key is used as defined in revision 1.3 or later of this specification. Bit 7 if cleared to '0' indicates that Ignore Existing Key is used as defined in revision 1.2.1 or earlier of this specification. This bit shall be set to '1' if the controller supports revision 1.3 or later as indicated in the Version register.</p> <p>Bit 6 if set to '1' indicates that the namespace supports the Exclusive Access – All Registrants reservation type. If this bit is cleared to '0', then the namespace does not support the Exclusive Access – All Registrants reservation type.</p> <p>Bit 5 if set to '1' indicates that the namespace supports the Write Exclusive – All Registrants reservation type. If this bit is cleared to '0', then the namespace does not support the Write Exclusive – All Registrants reservation type.</p> <p>Bit 4 if set to '1' indicates that the namespace supports the Exclusive Access – Registrants Only reservation type. If this bit is cleared to '0', then the namespace does not support the Exclusive Access – Registrants Only reservation type.</p> <p>Bit 3 if set to '1' indicates that the namespace supports the Write Exclusive – Registrants Only reservation type. If this bit is cleared to '0', then the namespace does not support the Write Exclusive – Registrants Only reservation type.</p> <p>Bit 2 if set to '1' indicates that the namespace supports the Exclusive Access reservation type. If this bit is cleared to '0', then the namespace does not support the Exclusive Access reservation type.</p> <p>Bit 1 if set to '1' indicates that the namespace supports the Write Exclusive reservation type. If this bit is cleared to '0', then the namespace does not support the Write Exclusive reservation type.</p> <p>Bit 0 if set to '1' indicates that the namespace supports the Persist Through Power Loss capability. If this bit is cleared to '0', then the namespace does not support the Persist Through Power Loss Capability.</p>
...

...

...

5.21.1.22 Host Behavior Support (Feature Identifier 16h)

...

Figure 317: Host Behavior Support – Data Structure

Bytes	Description
00	<p>Advanced Command Retry Enable (ACRE): If set to 1h, then the Command Interrupted status code is enabled (refer to Figure 128) and command retry delays are enabled. The controller may use the Command Interrupted status code and may indicate a command retry delay by setting the Command Retry Delay (CRD) field to a non-zero value in the Status field of a Completion Queue Entry, refer to Figure 126. A host that sets this field to 1h indicates host support for the command retry behaviors that are specified for both the Command Interrupted status code and non-zero values in the CRD field.</p> <p>If cleared to 0h, then both the Command Interrupted status code and command retry delays are disabled. The controller shall not use the Command Interrupted status code, and shall clear the CRD field to 0h in all CQEs.</p> <p>All values other than 0h and 1h are reserved.</p>

Bytes	Description
...	...
03	<p>Host Dispersed Namespace Support (HDISNS): If set to 1h, then dispersed namespaces are enabled. A host that sets this field to 1h specifies that globally unique namespace identifiers (refer to section 8.NEW.2) are used for identifying namespaces (e.g., multi-path I/O software does not use the NSID of a namespace as the sole method for detection of multiple paths to that namespace).</p> <p>A host that supports reservations (refer to section 8.8) and sets this field to 1h specifies a Host Identifier that is unique across all hosts that connect to any participating NVM subsystem.</p> <p>If cleared to 0h, then dispersed namespaces are disabled. If the participating NVM subsystem prohibits host access to dispersed namespaces (refer to section 8.NEW.3.1) when this field is cleared to 0h, then the controller aborts any I/O commands (refer to section 6) or any of the Admin commands listed in Figure NEW.7 that the host submits to dispersed namespaces with a status code of Host Dispersed Namespace Support Not Enabled, as described in section 8.NEW.3.1.</p> <p>All values other than 0h and 1h are reserved.</p>
511:0404	Reserved

...

6 NVM Command Set

...

6.4 Atomic Operations

...

In the case of a shared namespace (e.g., a dispersed namespace or shared namespace that is not a dispersed namespace), operations performed by an individual controller are atomic to the shared namespace at the write atomicity level reported in the corresponding Identify Controller or Identify Namespace data structures of the controller to which the command was submitted.

...

<5/26: FMDS group decided to remove all new requirements for write atomicity in section 6.4. Instead, a new paragraph was added to the end of this section to state why it's better to use the same write atomicity level across all controllers for dispersed namespaces (as a recommendation).>

<7/21: FMDS group decided to get rid of the recommendation paragraph entirely, as there could be a good reason why atomicity might differ between participating NVM subsystems (e.g., for the data migration use case). Now, the only time dispersed namespaces are mentioned with regards to atomicity is seen in the small addition to the existing paragraph in 6.4 shown above.>

...

6.10 Reservation Acquire command

...

Figure 382: Reservation Acquire – Command Dword 10

Bits	Description
31:16	Reserved
15:08	Reservation Type (RTYPE): This field specifies the type of reservation to be created. The field is defined in Figure 384.

Bits	Description										
07:0405	Reserved										
04	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>										
03	<p>Ignore Existing Key (IEKEY): If this bit is set to a '1', the controller shall return an error of Invalid Field In Command. If this bit is cleared to '0', then the Current Reservation Key is checked.</p>										
02:00	<p>Reservation Acquire Action (RACQA): This field specifies the action that is performed by the command.</p> <table border="1"> <thead> <tr> <th>RACQA Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Acquire</td> </tr> <tr> <td>001b</td> <td>Preempt</td> </tr> <tr> <td>010b</td> <td>Preempt and Abort</td> </tr> <tr> <td>011b to 111b</td> <td>Reserved</td> </tr> </tbody> </table>	RACQA Value	Description	000b	Acquire	001b	Preempt	010b	Preempt and Abort	011b to 111b	Reserved
RACQA Value	Description										
000b	Acquire										
001b	Preempt										
010b	Preempt and Abort										
011b to 111b	Reserved										

...

6.11 Reservation Register command

...

Figure 386: Reservation Register – Command Dword 10

Bits	Description										
31:30	<p>Change Persist Through Power Loss State (CPTPL): This field allows the Persist Through Power Loss (PTPL) state associated with the namespace to be modified as a side effect of processing this command. If the Reservation Persistence Feature (refer to section 5.21.1.28) is saveable, then any change to the PTPL state as a result of processing this command shall be applied to both the current value and the saved value of that feature.</p> <table border="1"> <thead> <tr> <th>CPTPL Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No change to PTPL state</td> </tr> <tr> <td>01b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>Set PTPL state to '0'. Reservations are released and registrants are cleared on a power on.</td> </tr> <tr> <td>11b</td> <td>Set PTPL state to '1'. Reservations and registrants persist across a power loss.</td> </tr> </tbody> </table>	CPTPL Value	Description	00b	No change to PTPL state	01b	Reserved	10b	Set PTPL state to '0'. Reservations are released and registrants are cleared on a power on.	11b	Set PTPL state to '1'. Reservations and registrants persist across a power loss.
CPTPL Value	Description										
00b	No change to PTPL state										
01b	Reserved										
10b	Set PTPL state to '0'. Reservations are released and registrants are cleared on a power on.										
11b	Set PTPL state to '1'. Reservations and registrants persist across a power loss.										

Bits	Description										
29:0405	Reserved										
04	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>										
03	<p>Ignore Existing Key (IEKEY): If this bit is set to a '1', then Reservation Register Action (RREGA) field values that use the Current Reservation Key (CRKEY) shall succeed regardless of the value of the Current Reservation Key field in the command (i.e., the current reservation key is not checked).</p>										
02:00	<p>Reservation Register Action (RREGA): This field specifies the registration action that is performed by the command.</p> <table border="1"> <thead> <tr> <th>RREGA Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Register Reservation Key</td> </tr> <tr> <td>001b</td> <td>Unregister Reservation Key</td> </tr> <tr> <td>010b</td> <td>Replace Reservation Key</td> </tr> <tr> <td>011b to 111b</td> <td>Reserved</td> </tr> </tbody> </table>	RREGA Value	Description	000b	Register Reservation Key	001b	Unregister Reservation Key	010b	Replace Reservation Key	011b to 111b	Reserved
RREGA Value	Description										
000b	Register Reservation Key										
001b	Unregister Reservation Key										
010b	Replace Reservation Key										
011b to 111b	Reserved										

...

6.12 Reservation Release command

...

Figure 389: Reservation Release – Command Dword 10

Bits	Description
31:16	Reserved
15:08	<p>Reservation Type (RTYPE): If the Reservation Release Action field is cleared to 000b (i.e., Release), then this field specifies the type of reservation that is being released. The reservation type in this field shall match the current reservation type. If the reservation type in this field does not match the current reservation type, then the controller should return an error of Invalid Field In Command. This field is defined in Figure 384.</p>
07:0405	Reserved

Bits	Description								
04	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>								
03	<p>Ignore Existing Key (IEKEY): If this bit is set to a '1', the controller shall return an error of Invalid Field in Command. If this bit is cleared to '0', then the Current Reservation Key is checked.</p>								
02:00	<p>Reservation Release Action (RRELA): This field specifies the registration action that is performed by the command.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Release</td> </tr> <tr> <td>001b</td> <td>Clear</td> </tr> <tr> <td>010b to 111b</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Description	000b	Release	001b	Clear	010b to 111b	Reserved
Value	Description								
000b	Release								
001b	Clear								
010b to 111b	Reserved								

...

6.13 Reservation Report command

The Reservation Report command returns a Reservation Status data structure to memory that describes the registration and reservation status of a namespace.

If the namespace is not a dispersed namespace, then the size of the Reservation Status data structure is a function of the number of controllers in the NVM subsystem that are associated with hosts that are registrants of the namespace (i.e., there is a Registered Controller data structure and/or Registered Controller extended data structure for each such controller). If the namespace is a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems, then the size of the Reservation Status data structure is a function of the number of controllers in the NVM subsystem containing the controller processing the command that are associated with hosts that are registrants of the namespace and the number of hosts that are registrants of the namespace in each separate participating NVM subsystem. The controller returns the data structure in Figure 394 if the host has selected a 64-bit Host Identifier and the data structure in Figure 395 if the host has selected a 128-bit Host Identifier (refer to section 5.21.1.26).

...

Figure 393: Reservation Report – Command Dword 11

Bits	Description
31:0402	Reserved

Bits	Description
01	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>
00	<p>Extended Data Structure (EDS): If set to '1', then the controller returns the extended data structure defined in Figure 395. If cleared to '0', then the controller returns the data structure defined in Figure 394.</p>

...

Figure 396: Registered Controller Data Structure

Bytes	Description
01:00	<p>Controller ID (CNTLID): This field contains the controller ID (i.e., the value of the CNTLID field in the Identify Controller data structure) of the controller whose status is reported in this data structure.</p> <p>If the controller is a dynamic controller (refer to the NVMe over Fabrics specification) that is not associated with a host, then the Controller ID field shall be set to FFFFh.</p> <p>If the namespace is a dispersed namespace and the controller is not contained in the same participating NVM subsystem as the controller processing the command, then the Controller ID field is set to FFFDh, as described in section 8.NEW.5.</p>
02	<p>Reservation Status (RCSTS): This field indicates the reservation status of the controller described by this data structure.</p> <p>Bits 7:1 are reserved</p> <p>Bit 0 is set to '1' if the controller is associated with a host that holds a reservation on the namespace.</p>
15:08	<p>Host Identifier (HOSTID): This field contains the 64-bit Host Identifier of the controller described by this data structure.</p>
23:16	<p>Reservation Key (RKEY): This field contains the reservation key of the host associated with the controller described by this data structure.</p>

...

7 Controller Architecture

...

7.1.1 I/O Controller

...

Figure 424: I/O Controller – Log Page Support

Log Page Name	Command Support Requirements ¹
...	...
Endurance Group Event Aggregate	O
Dispersed Namespace Participating NVM Subsystems	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited	

...

7.1.2 Administrative Controller

...

Figure 432: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements ¹
...	...
Endurance Group Event Aggregate	O
Dispersed Namespace Participating NVM Subsystems	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited	

...

...

7.11 Unique Identifier

<Move all existing text from 7.11 to new subsections (e.g., text pertaining to unique identifiers for NVM subsystems is moved into section 7.11.2 NVM Subsystem Unique Identifier).>

7.11.1 Unique Identifier Overview

Unique identifiers include the following:

- a) NVM subsystem unique identifiers (refer to section 7.11.2);
- b) controller unique identifiers (refer to section 7.11.3); and
- c) namespace unique identifiers (refer to section 7.11.4).

Unique identifiers are either globally unique or unique only within the NVM subsystem. NVM subsystem unique identifiers are globally unique. Controller unique identifiers (i.e., the CNTLID) are unique only within the NVM subsystem. Namespace unique identifiers are globally unique.

7.11.2 NVM Subsystem Unique Identifier

The NVM Subsystem NVMe Qualified Name specified in the Identify Controller data structure (refer to Figure 251) should be used (e.g., by host software) as the unique identifier for the NVM subsystem. If the controller complies with an older version of this specification that does not include the NVM Subsystem NQN, then the PCI Vendor ID, Serial Number, and Model Number fields in the Identify Controller data structure and the NQN Starting String “nqn.2014.08.org.nvmexpress:” may be combined by the host to form a globally unique value that identifies the NVM subsystem (e.g., for host software that uses NQNs). The method shown in Figure 452 should be used by the host to construct an NVM Subsystem NQN for older NVM subsystems that do not provide an NQN in the Identify Controller data structure. The mechanism used

by the vendor to assign Serial Number and Model Number values to ensure uniqueness is outside the scope of this specification.

Figure 452: NQN Construction for Older NVM Subsystems

Bytes	Description
26:00	NQN Starting String (NSS): Contains the 27 letter ASCII string "nqn.2014-08.org.nvmexpress:".
30:27	PCI Vendor ID (VID): Contains the company vendor identifier that is assigned by the PCI SIG as a hexadecimal ASCII string.
34:31	PCI Subsystem Vendor ID (SSVID): Contains the company vendor identifier that is assigned by the PCI SIG for the subsystem as a hexadecimal ASCII string.
54:35	Serial Number (SN): Contains the serial number for the NVM subsystem that is assigned by the vendor as an ASCII string.
94:55	Model Number (MN): Contains the model number for the NVM subsystem that is assigned by the vendor as an ASCII string.
222:95	Padding (PAD): Contains spaces (ASCII character 20h).

7.11.3 Controller Unique Identifier

An NVM subsystem may contain multiple controllers. All of the controllers that make up an NVM subsystem share the same NVM subsystem unique identifier. The Controller ID (CNTLID) value returned in the Identify Controller data structure may be used to uniquely identify a controller within an NVM subsystem. The Controller ID value when combined with the NVM subsystem identifier forms a globally unique value that identifies the controller. The mechanism used by the vendor to assign Controller ID values is outside the scope of this specification.

7.11.4 Namespace Unique Identifier

The Identify Namespace data structure (refer to Figure 249) contains the IEEE Extended Unique Identifier (EUI64) and the Namespace Globally Unique Identifier (NGUID) fields. The Namespace Identification Descriptor data structure (refer to Figure 253) contains the Namespace UUID. EUI64 is an 8-byte EUI-64 identifier (refer to section 7.10.4), NGUID is a 16-byte identifier based on EUI-64 (refer to section 7.10.5), and Namespace UUID is a 16-byte identifier described in RFC 4122 (refer to section 7.10.6).

When creating a namespace, the controller shall indicate a globally unique ~~value~~ namespace identifier in one or more of the following:

- a) the EUI64 field;
- b) the NGUID field; or
- c) a Namespace Identification Descriptor with the Namespace Identifier Type field set to 3h (i.e., a UUID).

Refer to section [8.NEW.2](#) for additional globally unique namespace identifier requirements related to dispersed namespaces.

If the EUI64 field is cleared to 0h and the NGUID field is cleared to 0h, then the namespace shall support a valid Namespace UUID in the Namespace Identification Descriptor data structure.

If the UIDREUSE bit in the NSFEAT field is cleared to '0', then a controller may reuse a non-zero NGUID/EUI64 value for a new namespace after the original namespace using the value has been deleted. If the UIDREUSE bit is set to '1', then a controller shall not reuse a non-zero NGUID/EUI64 for a new namespace after the original namespace using the value has been deleted.

8 Features

...

8.8 Reservations (Optional)

...

Controllers that make up an NVM subsystem shall all have the same support for reservations. Although strongly encouraged, namespaces that make up an NVM subsystem are not all required to have the same support for reservations. For example, some namespaces within a single controller may support reservations while others do not, or the supported reservation types may differ among namespaces. If a controller supports reservations, then the controller shall:

- Indicate support for reservations by returning a '1' in bit 5 of the Optional NVM Command Support (ONCS) field in the Identify Controller data structure;
- Support the Reservation Report command (refer to section 6.13), Reservation Register command (refer to section 6.11), Reservation Acquire command (refer to section 6.10), and Reservation Release command (refer to section 6.12);
- Support the Reservation Notification log page;
- Support the Reservation Log Page Available asynchronous events;
- Support the Reservation Notification Mask Feature;
- Support the Host Identifier Feature; and
- Support the Reservation Persistence Feature.

If a controller supports dispersed namespaces and supports reservations, then the controller supports the Dispersed Namespace Reservation Support (DISNSRS) bit being set to '1' in the Reservation Report command, Reservation Register command, Reservation Acquire command, and Reservation Release command, as described in section [8.NEW.5](#).

If a namespace supports reservations, then the namespace shall:

- Report a non-zero value in the Reservation Capabilities (RESCAP) field in the Identify Namespace data structure;
- Support Persist Through Power Loss (PTPL) state; and
- Support sufficient resources to allow a host to successfully register a reservation key on every controller in the NVM subsystem with access to the shared namespace (i.e., a Reservation Register command shall never fail due to lack of resources).

...

...

8.8.2 Registering

...

Registering a reservation key with a namespace creates an association between a host and a namespace. A host that is a registrant of a namespace may use any controller with which that host is associated (i.e., that has the same Host Identifier, refer to section 5.21.1.26) to access that namespace as a registrant. Thus, a host is only required to register on a single controller to become a registrant of the namespace on all controllers in the NVM subsystem that have access to the namespace and are associated with the host. If a host attempts to access a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems, then that host is only required to register on a single controller in a single participating NVM subsystem to become a registrant of the dispersed namespace on all controllers in all participating NVM subsystems that have access to the dispersed namespace and are associated with the host.

...

A host that is a registrant of a namespace may replace the existing reservation key value for that namespace by executing a Reservation Register command on the namespace with the:

- a) RREGA field set to 010b (i.e., Replace Reservation Key);
- b) current reservation key in the Current Reservation Key (CRKEY) field; and
- c) new reservation key in the NRKEY field.

The current reservation key value shall be replaced by the new reservation key value in all controllers to which the namespace is attached that have the same Host Identifier as the Host Identifier of the controller processing the command. [For dispersed namespaces, this requirement includes all participating NVM subsystems.](#) If the contents of the CRKEY field do not match the key currently associated with the host, then the command shall be aborted with a status of Reservation Conflict. A host may replace its reservation key without regard to its registration status or current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the Reservation Register command. Replacing a reservation key has no effect on any reservation that may be held on the namespace.

...

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). The method for assigning namespaces to ANA Groups is outside the scope of this specification.

A valid ANA Group Identifier is a non-zero value that is less than or equal to ANAGRPMAX (refer to Figure 251).

The ANA Group Identifier (ANAGRPID) for each ANA Group shall be unique within the NVM subsystem. If bit 6 in the ANACAP field in the Identify Controller data structure is set to '1', then the ANA Group Identifier shall not change while the namespace is attached to any controller in the NVM subsystem. If bit 6 in the ANACAP field is cleared to '0', then the ANA Group Identifier may change while the namespace is attached to any controller in the NVM subsystem. If the ANA Group Identifier changes, the controller shall issue the Asymmetric Namespace Access Change Notice as described in 8.20.3.6.

[If two or more participating NVM subsystems provide access to a dispersed namespace, and that dispersed namespace is a member of an ANA Group on any participating NVM subsystem, then each participating NVM subsystem shall contain an ANA Group whose members include that dispersed namespace. The ANAGRPID for a dispersed namespace may or may not be the same on each participating NVM subsystem.](#)

[<ANA grouping requirements about each participating NVM subsystem maintaining the same ANA Group membership for overlapping sets of dispersed namespaces has been removed. For example, it's now permissible to have ANA Group 1 on NVM Subsystem 1 contain dispersed namespaces A and B together and for ANA Group 2 on NVM Subsystem 2 to contain only dispersed namespace A and for ANA Group 3 on NVM Subsystem 2 to contain only dispersed namespace B.>](#)

...

...

8.21.1 Host ANA Normal Operation

The host determines if ANA is supported by examining bit 3 in the CMIC field in the Identify Controller data structure (refer to Figure 251). The NSID or Identifier (refer to section 7.10) is used to determine when multiple paths to the same namespace [that is not a dispersed namespace](#) are available. [For dispersed namespaces, globally unique namespace identifiers \(refer to section 8.NEW.2\) are used to determine when multiple paths are available to the same namespace, as described in section 8.NEW.4.](#) The host examines

the ANA Log page (refer to section 5.14.1.12) for each controller to determine the ANA state of each group of namespaces attached to that controller.

...

8. NEW Dispersed Namespaces (Optional)

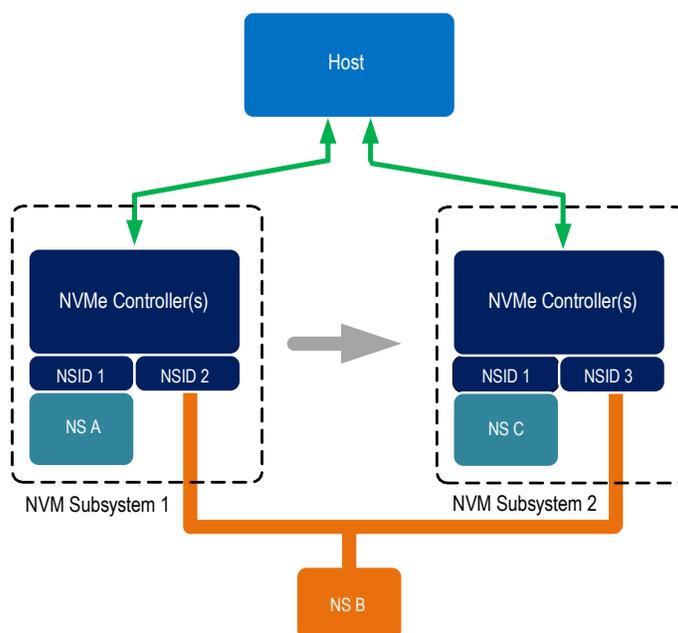
<This will be the model clause.>

Dispersed namespaces provide hosts with access to the same namespace using multiple participating NVM subsystems. Two prominent scenarios that require namespace access from multiple participating NVM subsystems are:

1. online data migration; and
2. data replication.

Online data migration is used to transparently move the contents of one or more namespaces across participating NVM subsystems without disrupting host access to those namespaces during the migration. For each namespace whose contents are migrated, one or more paths to the namespace in the destination NVM subsystem are added to the multi-path I/O on the host, in addition to the existing paths to the namespace in the source NVM subsystem. Once the online data migration completes (i.e., all of the contents of all namespaces being migrated has been successfully migrated from the source NVM subsystem to the destination NVM subsystem), all of the paths to the namespace in the source NVM subsystem are subsequently removed from the multi-path I/O on the host. During the online data migration, the host and participating NVM subsystems must manage the paths so that there is no interruption to I/O. Figure NEW.3 shows an example of online data migration, where the contents of namespace B are migrated from one participating NVM subsystem to another participating NVM subsystem. In this figure, the lines from the host to the NVMe controllers may represent multiple connections from the host to multiple controllers in each participating NVM subsystem.

Figure NEW.3: Online Data Migration



Data replication is used to replicate the contents of one or more dispersed namespaces across participating NVM subsystems such that each participating NVM subsystem is able to provide host access to the same namespace data. Other behavior (e.g., persistent reservations) must also be

maintained between hosts and each participating NVM subsystem. Figure NEW.4, Figure NEW.5, and Figure NEW.6 show different examples of data replication, where the contents of namespace B are replicated across two participating NVM subsystems. In these figures, the lines from hosts to NVMe controllers may represent multiple connections from the hosts to multiple controllers in each participating NVM subsystem. Figure NEW.4 shows an example of single NVM subsystem host connectivity, where two hosts access the same dispersed namespace being replicated across two participating NVM subsystems via controllers in only one NVM subsystem.

Figure NEW.4: Data Replication Example 1

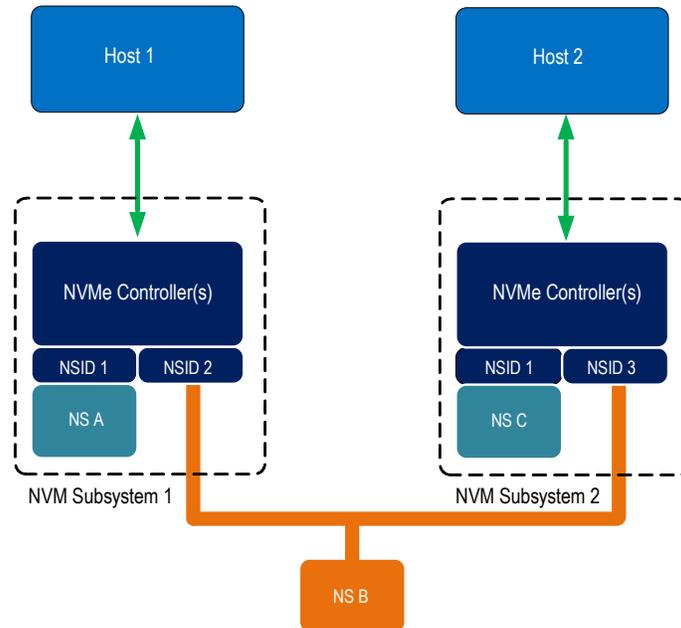


Figure NEW.5 shows another example of single NVM subsystem host connectivity, where two hosts access the same dispersed namespace being replicated across two participating NVM subsystems via controllers in only one NVM subsystem, but the host connected to NVM Subsystem 2 only accesses namespace B in the event of a failure scenario.

Figure NEW.5: Data Replication Example 2

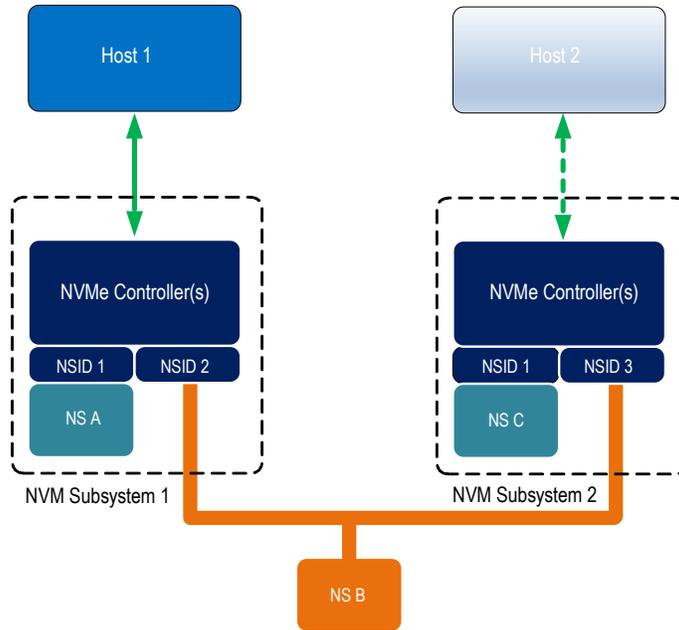
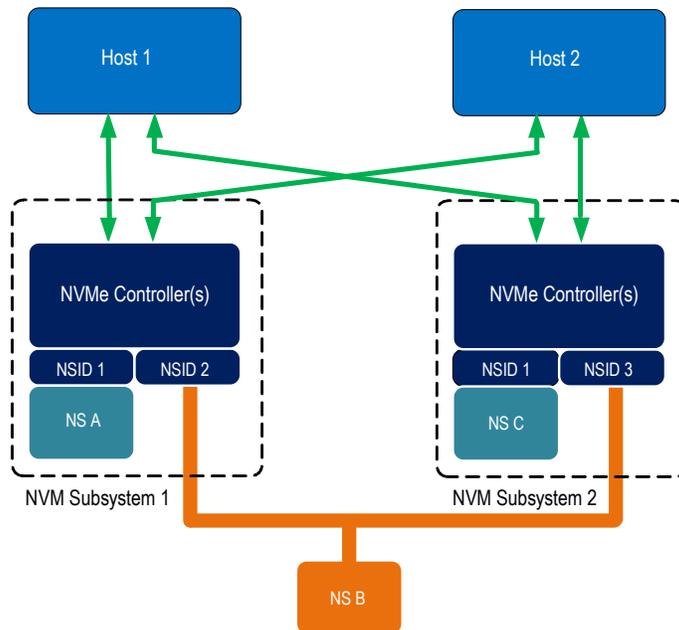


Figure NEW.6 shows an example of multiple NVM subsystem host connectivity, where two hosts access the same dispersed namespace being replicated across two participating NVM subsystem via controllers in both NVM subsystems.

Figure NEW.6: Data Replication Example 3



8.NEW.1 Dispersed Namespace Management

The Namespace Management command (refer to section 5.20) may be used to create a namespace that is capable of being accessed using controllers contained in two or more participating NVM subsystems concurrently (i.e., may be used to create a namespace that is capable of being a dispersed namespace). The methods for dispersing a namespace (e.g., attaching that namespace to controllers contained in separate participating NVM subsystems) are outside the scope of this specification. The Namespace Management or Capacity Management command may be used to delete a dispersed namespace on the participating NVM subsystem containing the controller processing the command. Deleting a dispersed namespace on one participating NVM subsystem may or may not affect that namespace on other participating NVM subsystems (e.g., a deletion from one participating NVM subsystem may have no effect on the namespace on other participating NVM subsystems, or a deletion from one participating NVM subsystem may delete the namespace from all participating NVM subsystems).

The Namespace Management command may be used to create a shared namespace that is not a dispersed namespace which is later converted to a dispersed namespace. The method of converting a namespace that is not a dispersed namespace to a dispersed namespace is outside the scope of this specification. NVM subsystems should not convert private namespaces to dispersed namespaces. Whenever a namespace that is not a dispersed namespace is converted to a dispersed namespace or a dispersed namespace is converted to a namespace that is not a dispersed namespace, the controller reports a Namespace Attribute Changed event as described in Figure 149.

The host may attach or detach a dispersed namespace from the controller by using the Namespace Attachment command (refer to section 5.19).

8.NEW.2 NSID and Globally Unique Namespace Identifier Usage

The NSID for a dispersed namespace is unique for all controllers in a participating NVM subsystem, as described in section 6.1.6. The NSID for a dispersed namespace may be different on different participating NVM subsystems.

A dispersed namespace has a globally unique namespace identifier (refer to section 7.11.4) that uniquely identifies the namespace in all participating NVM subsystems. The globally unique namespace identifier has the same value for that namespace in all participating NVM subsystems and shall be used to determine which NSIDs on different participating NVM subsystems refer to the same namespace. The globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems shall be either:

- a) a Namespace Globally Unique Identifier (NGUID); or
- b) a Universally Unique Identifier (UUID).

If the dispersed namespace has an NGUID and does not have a UUID, then:

- a) that NGUID shall be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems;
- b) all participating NVM subsystems shall use the same NGUID value for the dispersed namespace; and
- c) the dispersed namespace may have an EUI64 that shall not be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems.

If the dispersed namespace has a UUID and does not have an NGUID, then:

- a) that UUID shall be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems;
- b) all participating NVM subsystems shall use the same UUID value for the dispersed namespace; and
- c) the dispersed namespace may have an EUI64 that shall not be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems.

If the dispersed namespace has an NGUID and has a UUID, then:

- a) the NGUID shall be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems;
- b) all participating NVM subsystems shall use the same NGUID value for the dispersed namespace; and
- c) the dispersed namespace may have an EUI64 that shall not be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems.

8.NEW.3 Dispersed Namespace Access

The host may discover if a namespace is capable of being accessed using controllers contained in two or more participating NVM subsystems concurrently (i.e., may discover if a namespace may be a dispersed namespace) by issuing an Identify command (i.e., CNS 00h) to the controller for the specified NSID. If the namespace is capable of being accessed using controllers contained in two or more participating NVM subsystem concurrently, then the controller shall set the Dispersed Namespace (DISNS) bit to '1' in the Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC) field of the returned Identify Namespace data structure (refer to Figure 249).

All controllers in each participating NVM subsystem that are able to provide access to a specific dispersed namespace shall support the I/O Command Set associated with that dispersed namespace.

The host may discover the NQNs of participating NVM subsystems which contain controllers that are able to provide access to a dispersed namespace by issuing a Get Log Page command with the Log Page Identifier (LID) field set to 17h (refer to section **5.14.1.NEW**).

Hosts specify support for dispersed namespaces by setting the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature by issuing a Set Features command with Feature Identifier (FID) set to 16h (refer to Figure 317).

If the HDISNS field is set to 1h, the host specifies support for:

- treating all instances of a dispersed namespace as the same namespace, based upon the shared globally unique namespace identifier (refer to section **8.NEW.2**) when accessing that dispersed namespace through multiple controllers on the same participating NVM subsystem;
- treating all instances of a dispersed namespace as the same namespace, based upon the shared globally unique namespace identifier when accessing that dispersed namespace through controllers on multiple participating NVM subsystems; and
- specifying a Host Identifier that is unique across all hosts that connect to any participating NVM subsystem.

If the HDISNS field is set to 1h, the controller shall not abort any command that the host submits to dispersed namespaces with a status code of Host Dispersed Namespace Support Not Enabled.

<There could be corner cases where an old host (HDISNS = 0) will get RESERVATION CONFLICT but not be able to find out why if it can't get back CNTLID = FFFDh. After consulting with a number of host developers, it was determined that getting back FFFDh will not be a problem, so the requirement that HDISNS = 1 before controllers can report CNTLID = FFFDh has been removed.>

A participating NVM subsystem shall support prohibiting host access to dispersed namespaces when the HDISNS field is cleared to 0h in the Host Behavior Support feature, as described in section **8.NEW.3.1**. A participating NVM subsystem may either:

- allow host access to a dispersed namespace when the HDISNS field is cleared to 0h in the Host Behavior Support feature if the host associated with the Host NQN in the connection (refer to the NVMe over Fabrics specification) is only able to access the dispersed namespace from a single participating NVM subsystem; or
- prohibit host access to dispersed namespaces when the HDISNS field is cleared to 0h in the Host Behavior Support feature.

A participating NVM subsystem that allows host access to a dispersed namespace when the HDISNS field is cleared to 0h in the Host Behavior Support feature if the host associated with the Host NQN in the connection is only able to access the dispersed namespace from a single participating NVM subsystem shall implement a method to determine if the host is able to access the dispersed namespace from multiple participating NVM subsystems. The method used to determine if the host is able to access the dispersed namespace from multiple participating NVM subsystems is outside the scope of this specification. If a participating NVM subsystem determines that a Connect command requests a host with the HDISNS field cleared to 0h in the Host Behavior Support feature having access to the same dispersed namespace through multiple participating NVM subsystems, then that participating NVM subsystem shall either:

- a) abort a Connect command with a status code of Connect Invalid Host (refer to the NVMe over Fabrics specification) if that Connect command requests the host associated with the Host NQN in the connection having access to a dispersed namespace through multiple participating NVM subsystems. Aborting a Connect command for this reason may result in the host being unable to access other namespaces (i.e., namespaces that are unrelated to the dispersed namespace) on that NVM subsystem; or
- b) allow a Connect command that requests the host associated with the Host NQN in the connection having access to a dispersed namespace through multiple participating NVM subsystems, but prohibit host access to that dispersed namespace from multiple participating NVM subsystems (e.g., prohibit host access to that dispersed namespace on the second participating NVM subsystem) as described in section **8.NEW.3.1**.

Option a) in the preceding list is intended to enforce the requirement that a host that has not set the HDISNS field to 1h in the Host Behavior Support feature be allowed to access a dispersed namespace on only one participating NVM subsystem by preventing all access to any other participating NVM subsystems (including all other namespaces on those other participating NVM subsystems). Option b) in the preceding list is intended to enforce the requirement that a host that has not set the HDISNS field to 1h in the Host Behavior Support feature be allowed to access a dispersed namespace on only one participating NVM subsystem by, for each dispersed namespace, preventing access to that dispersed namespace on any other participating NVM subsystems (and allowing access to other namespaces that are not dispersed namespaces on those other participating NVM subsystems).

8.NEW.3.1 Prohibiting Host Access to Dispersed Namespaces

If a participating NVM subsystem prohibits host access to dispersed namespaces when the Host Dispersed Namespace Support (HDISNS) field is cleared to 0h in the Host Behavior Support feature, then the controller shall abort the following commands with a status code of Host Dispersed Namespace Support Not Enabled:

- any I/O commands (refer to section 6) that specify the NSID of a dispersed namespace in the command; or
- any of the Admin commands listed in Figure **NEW.7** that specify the NSID of a dispersed namespace in the command.

The Additional Restrictions column in Figure **NEW.7** lists additional restrictions that apply while the HDISNS field is cleared to 0h in the Host Behavior Support feature for Admin commands that are capable of affecting dispersed namespaces without specifying the NSID of a dispersed namespace.

Figure **NEW.7: Dispersed Namespaces Command Restrictions – Prohibited Admin Commands**

Admin Command	Additional Restrictions
Directive Receive	None.
Directive Send	None.

Figure NEW.7: Dispersed Namespaces Command Restrictions – Prohibited Admin Commands

Admin Command	Additional Restrictions
Set Features	This command is prohibited if a dispersed namespace is attached to the controller and an NSID of FFFFFFFFh is specified for a feature capable of affecting all namespaces attached to that controller.
Format NVM	<p>This command is prohibited if a dispersed namespace is allocated in the participating NVM subsystem and the scope of the command includes all allocated namespaces in that NVM subsystem (refer to Figure 331).</p> <p>This command is prohibited if a dispersed namespace is attached to the controller and the scope of the command includes all namespaces attached to that controller (refer to Figure 331).</p>

8.NEW.4 ANA Considerations

If more than one participating NVM subsystem contains controllers that provide a host with access to a dispersed namespace, then Asymmetric Namespace Access (ANA) and Asymmetric Namespace Access Reporting (refer to section 8.20) should be supported in all participating NVM subsystems. If ANA is used with dispersed namespaces, then globally unique namespace identifiers (refer to section 8.NEW.2) are used to determine when multiple paths are available to the same dispersed namespace. Hosts specify to controllers that their host software (e.g., multipath I/O software) uses globally unique namespace identifiers to determine when multiple paths are available to the dispersed namespace by setting the HDISNS field to 1h in the Host Behavior Support feature, as described in section 8.NEW.3.

ANA Group usage with dispersed namespaces is described in section 8.20.2.

8.NEW.5 Reservation Considerations

Reservations for a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems are intended to be coordinated between each participating NVM subsystem. To ensure the uniqueness of a host's identity and prevent potential data corruption, each host that sets the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature specifies a Host Identifier that is unique across all hosts that connect to any participating NVM subsystem, as described in section 8.NEW.3.

If a host is a reservation registrant on a dispersed namespace, and that host submits a Reservation Report command to that namespace, then for reservations established using controllers that are not contained in the same participating NVM subsystem as the controller processing the Reservation Report command, the Controller ID (CNTLID) field shall be set to FFFDh in the Registered Controller data structure (refer to Figure 396) or Registered Controller Extended data structure (refer to Figure 397). Multiple identical Registered Controller or Registered Controller Extended data structures with the CNTLID field set to FFFDh shall not be returned by a single Reservation Report command; as a result, a Registered Controller or Registered Controller Extended data structure with the CNTLID field set to FFFDh indicates one or more controllers are associated with a host that is a registrant of the dispersed namespace in another participating NVM subsystem. A Registered Controller or Registered Controller Extended data structure with the CNTLID field set to FFFDh is returned for each host (i.e., as identified by the HOSTID field in that data structure) that is a registrant of the dispersed namespace in any other participating NVM subsystem.

The Dispersed Namespace Reservation Support (DISNSRS) bit in a command is set to '1' by hosts that support reservations on dispersed namespaces (i.e., hosts that support receiving a value of FFFDh in the CNTLID field of a Registered Controller data structure or Registered Controller Extended data structure). A controller that supports dispersed namespaces and supports reservations (i.e., a controller that has Bit 5 set to '1' in the Optional NVM Command Support (ONCS) field of the Identify Controller data structure (refer to Figure 251)) shall support the DISNSRS bit being set to '1' in each command in the following list:

- Reservation Acquire (refer to section 6.10);
- Reservation Register (refer to section 6.11);
- Reservation Release (refer to section 6.12); and
- Reservation Report (refer to section 6.13).

If the HDISNS field is set to 1h in the Host Behavior Support feature and the DISNSRS bit is cleared to '0' in any of the commands in the preceding list when submitted to a dispersed namespace, then the controller shall abort that command with a status code of Namespace Is Dispersed.

Description of NVMe-oF 1.1a Specification Changes

3 Commands

...

3.3 Connect Command and Response

...

The NVM subsystem shall not allocate a Controller ID in the range FFF0h to FFFFh as a valid Controller ID on completion of a Connect command. Controller IDs FFF0h to FFFFh are defined in Figure NEW.A. If the host is not allowed to establish an association to any controller in the NVM subsystem, then a status of Connect Invalid Host is returned.

...

A status of Connect Invalid Parameters is returned for a Connect command if:

- the host sends a Connect command to create an I/O Queue while the controller is disabled;
- the Host Identifier, Host NQN, NVM Subsystem NQN, and the Controller ID values specified for an I/O Queue are not the same as the values specified for the associated Admin Queue in which the association between the host and controller was established;
- the Host NQN or NVM Subsystem NQN values do not match the values that the NVM subsystem is configured to support;
- there is a syntax error in the Host NQN or NVM Subsystem NQN value (refer to section 7.9 in the NVMe Base specification); ~~or~~
- the Host Identifier is cleared to 0h; ~~or~~
- the host specifies a Controller ID value in the range FFF0h to FFFDh.

...

4 Controller Architecture

...

4.2 Controller Model

The NVM subsystem may support a dynamic or static controller model. All controllers in the NVM subsystem shall follow the same controller model. A Discovery Controller shall support the dynamic controller model.

Controller IDs in the range FFF0h to FFFFh are not allocated as valid Controller IDs on completion of a Connect command, as described in section 3.3. Figure NEW.A defines these Controller IDs.

Figure NEW.A: Controller IDs FFF0h to FFFFh

Controller ID	Definition
FFF0h to FFFCh	Reserved. Use of this value in a Connect command results in a status code of Connect Invalid Parameters being returned, as described in section 3.3.
FFFDh	This value in the Controller ID (CNTLID) field of the Registered Controller data structure or Registered Controller Extended data structure for a dispersed namespace indicates that the controller is not contained in the same participating NVM subsystem as the controller processing the command (refer to section 8.NEW.5 in the NVMe Base Specification). Use of this value in a Connect command results in a status code of Connect Invalid Parameters being returned, as described in section 3.3.
FFFEh	This value is sent in a Connect command to specify that any available static controller is allowed to be allocated.
FFFFh	This value is sent in a Connect command to specify that any available dynamic controller is allowed to be allocated.

...

The controller ID values returned in the Discover Log entries indicate whether an NVM subsystem supports the dynamic or static controller model. The controller ID value of FFFFh is ~~a special value~~ used for NVM subsystems that support the dynamic controller model indicating that any available controller may be returned. The controller ID value of FFFEh is ~~a special value~~ used for NVM subsystems that support the static controller model indicating that any available controller may be returned. An NVM subsystem supports the dynamic controller model if Discovery Log entries use the Controller ID value of FFFFh. An NVM subsystem supports the static controller model if Discovery Log entries use a Controller ID value that is less than FFFFh. The Identify Controller data structure also indicates whether an NVM subsystem is dynamic or static.

...

Description of NVMe-MI 1.2 Specification Changes

6 NVM Express Admin Command Set

...

6.3 Get Log Page

...

Figure 121: Management Endpoint - Log Page Support

Log Page Name ³	Log Identifier	SMBus/I2C Log Page Support Requirements ¹		PCIe VDM Log Page Support Requirements ¹	
		NVMe Storage Device	NVMe Enclosure	NVMe Storage Device	NVMe Enclosure
...
Rotational Media Information	16h	O	O	O	O
Dispersed Namespace Participating NVM Subsystems	17h	O	O	O	O
Discovery	70h	O	O	O	O
...

Notes:

1. O = Optional, M = Mandatory, P = Prohibited.
2. Optional for versions 1.1 and earlier of this specification.
3. Refer to the NVM Express Base Specification unless another footnote specifies otherwise.
4. Refer to the NVM Command Set Specification.
5. Refer to the Zoned Namespace Command Set Specification.

Description of NVMe 2.0 Base Specification Changes

1 Introduction

...

1.5 Definitions

...

1.5.NEW1 dispersed namespace

A shared namespace that may be concurrently accessed by controllers in two or more NVM subsystems (refer to section 8.NEW).

1.5.NEW2 participating NVM subsystem

An NVM subsystem that participates in (i.e., contains controllers that provide access to) a dispersed namespace.

3 NVM Express Architecture

...

3.1.2.1 I/O Controller

...

3.1.2.1.2 Log Page Support

...

Figure 24: I/O Controller – Log Page Support

Log Page Name	Log Page Support Requirements ¹
...	...
Rotational Media Information	O
Dispersed Namespace Participating NVM Subsystems	O
Notes: 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited 2. Mandatory for controllers that support Fixed Capacity Management (refer to section 8.3.2). 3. Optional for NVM Express revision 1.4 and earlier.	

...

3.1.2.2 Administrative Controller

...

3.1.2.2.2 Log Page Support

...

Figure 29: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements ¹
...	...

Figure 29: Administrative Controller – Log Page Support

Log Page Name	Command Support Requirements ¹
I/O Command Set Specific Log Pages	P
Dispersed Namespace Participating NVM Subsystems	O
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.	

...

3.1.2.3 Discovery Controller

...

Multiple Discovery Log Page Entries for the same NVM subsystem with different Port ID values indicates that the resulting NVMe Transport connections are independent with respect to NVM subsystem port hardware failures. A host that uses a single association should pick a record to attach to an NVM subsystem. A host that uses multiple associations should choose different ports.

A transport specific method may exist to indicate changes to a Discovery controller.

Controller IDs in the range FFF0h to FFFFh are not allocated as valid Controller IDs on completion of a Connect command, as described in section 6.3. Figure **NEW.A** defines these Controller IDs.

Figure NEW.A: Controller IDs FFF0h to FFFFh

Controller ID	Definition
FFF0h to FFFCh	Reserved. Use of this value in a Connect command results in a status code of Connect Invalid Parameters being returned, as described in section 6.3.
FFFDh	This value in the Controller ID (CNTLID) field of the Registered Controller data structure or Registered Controller Extended data structure for a dispersed namespace indicates that the controller is not contained in the same participating NVM subsystem as the controller processing the command (refer to section 8.NEW.5). Use of this value in a Connect command results in a status code of Connect Invalid Parameters being returned, as described in section 6.3.
FFFEh	This value is sent in a Connect command to specify that any available static controller is allowed to be allocated.
FFFFh	This value is sent in a Connect command to specify that any available dynamic controller is allowed to be allocated.

The Controller ID values returned in the Discovery Log Page Entries indicate whether an NVM subsystem supports the dynamic or static controller model. The controller ID value of FFFFh is ~~a special value~~ used for NVM subsystems that support the dynamic controller model indicating that any available controller may be returned. The Controller ID value of FFFEh is ~~a special value~~ used for NVM subsystems that support the static controller model indicating that any available controller may be returned. An NVM subsystem supports the dynamic controller model if Discovery Log Page Entries use the Controller ID value of FFFFh. An NVM subsystem supports the static controller model if Discovery Log Page Entries use a Controller ID value that is less than FFFFh. The Identify Controller data structure also indicates whether an NVM subsystem is dynamic or static.

...

...

3.1.2.3.3 Log Page Support

...

Figure 33: Discovery Controller – Log Page Support

Log Page Name	Command Support Requirements ¹
...	...
I/O Command Set Specific Log Pages	P
Dispersed Namespace Participating NVM Subsystems	P
Notes 1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited 2. Optional if Set Features command is not supported (refer to Figure 32). 3. Optional if NVMe-MI Send command and NVMe-MI Receive command is not supported (refer to Figure 32). 4. Optional for versions 1.1 and earlier of the NVMe over Fabrics specification.	

...

3.3.3.2.1.1 Generic Command Status Definition

...

Figure 94: Status Code – Generic Command Status Values

Value	Description	I/O Command Set Specific	I/O Command Set(s) ¹
...
24h	<p>Admin Command Media Not Ready: The Admin command requires access to media and the media is not ready. The Do Not Retry bit indicates whether re-issuing the command at a later time may succeed. This status code shall only be returned:</p> <ul style="list-style-type: none"> a) for Admin commands; and b) if the controller is in Controller Ready Independent of Media mode (CC.CRIME is bit is set to '1'). <p>This status code shall not be returned with the Do Not Retry bit cleared to '0' after the amount of time indicated by the Controller Ready With Media Timeout field (CRTO.CRWMT) after the controller is enabled (i.e., CC.EN transitions from '0' to '1').</p> <p>Refer to Figure 103 for the list of Admin commands permitted to return this status code.</p>	No	
...
26h	<p>Host Dispersed Namespace Support Not Enabled: The command is prohibited while the Host Dispersed Namespace Support (HDISNS) field is not set to 1h in the Host Behavior Support feature (refer to Figure 351).</p>	No	
25h 27h to 7Fh	Reserved		
80h	<p>LBA Out of Range: See the applicable I/O Command Set specification for the description.</p>	Yes	NVM, ZNS
...

Figure 94: Status Code – Generic Command Status Values

Value	Description	I/O Command Set Specific	I/O Command Set(s) ¹
Key: NVM – NVM Command Set ZNS – Zoned Namespace Command Set KV – Key Value Command Set NOTES: 1. This column is blank unless the value is I/O Command Set specific			

3.3.3.2.1.2 Command Specific Status Definition

...

Figure 95: Status Code – Command Specific Status Values

Value	Description	Commands Affected
...
2Dh	Identifier Unavailable	Capacity Management
2Eh	Namespace Is Dispersed	Reservation Acquire, Reservation Register, Reservation Release, Reservation Report
2Eh 2Fh to 6Fh	Reserved	
70h to 7Fh	Directive Specific	NOTE 1
...
NOTES: 1. The Directives Specific range defines Directives specific status values. Refer to section 8.7.		

...

4 Data Structures

...

4.5.1 Unique Identifier

<Move all existing text from 4.5.1 to new subsections (e.g., text pertaining to unique identifiers for NVM subsystems is moved into section 4.5.1.2 NVM Subsystem Unique Identifier).>

4.5.1.1 Unique Identifier Overview

Unique identifiers include the following:

- a) NVM subsystem unique identifiers (refer to section 4.5.1.2);
- b) controller unique identifiers (refer to section 4.5.1.3); and
- c) namespace unique identifiers (refer to section 4.5.1.4).

Unique identifiers are either globally unique or unique only within the NVM subsystem. NVM subsystem unique identifiers are globally unique. Controller unique identifiers (i.e., the CNTLID) are unique only within the NVM subsystem. Namespace unique identifiers are globally unique.

4.5.1.2 NVM Subsystem Unique Identifier

The NVM Subsystem NVMe Qualified Name specified in the Identify Controller data structure (refer to Figure 275) should be used (e.g., by host software) as the unique identifier for the NVM subsystem. If the controller complies with an older version of the NVM Express specification that does not include the NVM

Subsystem NQN, then the PCI Vendor ID, Serial Number, and Model Number fields in the Identify Controller data structure and the NQN Starting String “nqn.2014.08.org.nvmexpress:” may be combined by the host to form a globally unique value that identifies the NVM subsystem (e.g., for host software that uses NQNs). The method shown in Figure 137 should be used by the host to construct an NVM Subsystem NQN for older NVM subsystems that do not provide an NQN in the Identify Controller data structure. The mechanism used by the vendor to assign Serial Number and Model Number values to ensure uniqueness is outside the scope of this specification.

Figure 137: NQN Construction for Older NVM Subsystems

Bytes	Description
26:00	NQN Starting String (NSS): Contains the 27 letter ASCII string “nqn.2014-08.org.nvmexpress:”.
30:27	PCI Vendor ID (VID): Contains the company vendor identifier that is assigned by the PCI SIG as a hexadecimal ASCII string.
34:31	PCI Subsystem Vendor ID (SSVID): Contains the company vendor identifier that is assigned by the PCI SIG for the subsystem as a hexadecimal ASCII string.
54:35	Serial Number (SN): Contains the serial number for the NVM subsystem that is assigned by the vendor as an ASCII string.
94:55	Model Number (MN): Contains the model number for the NVM subsystem that is assigned by the vendor as an ASCII string.
222:95	Padding (PAD): Contains spaces (ASCII character 20h).

4.5.1.3 Controller Unique Identifier

An NVM subsystem may contain multiple controllers. All controllers contained in the NVM subsystem share the same NVM subsystem unique identifier. The Controller ID (CNTLID) value returned in the Identify Controller data structure may be used to uniquely identify a controller within an NVM subsystem. The Controller ID value when combined with the NVM subsystem identifier forms a globally unique value that identifies the controller. The mechanism used by the vendor to assign Controller ID values is outside the scope of this specification.

4.5.1.4 Namespace Unique Identifier

The Identify Namespace data structure (refer to the applicable I/O Command Set specification) contains the IEEE Extended Unique Identifier (EUI64) and the Namespace Globally Unique Identifier (NGUID) fields. The Namespace Identification Descriptor data structure (refer to Figure 277) contains the Namespace UUID. EUI64 is an 8-byte EUI-64 identifier (refer to section 4.3.4), NGUID is a 16-byte identifier based on EUI-64 (refer to section 4.3.5), and Namespace UUID is a 16-byte identifier described in RFC 4122 (refer to section 4.3.6).

When creating a namespace, the controller shall indicate a globally unique ~~value~~namespace identifier in one or more of the following:

- a) the EUI64 field;
- b) the NGUID field; or
- c) a Namespace Identification Descriptor with the Namespace Identifier Type field set to 3h (i.e., a UUID).

Refer to section [8.NEW.2](#) for additional globally unique namespace identifier requirements related to dispersed namespaces.

If the EUI64 field is cleared to 0h and the NGUID field is cleared to 0h, then the namespace shall support a valid Namespace UUID in the Namespace Identification Descriptor data structure.

If the UIDREUSE bit in the NSFEAT field is cleared to ‘0’, then a controller may reuse a non-zero NGUID/EUI64 value for a new namespace after the original namespace using the value has been deleted.

If the UIDREUSE bit is set to '1', then a controller shall not reuse a non-zero NGUID/EUI64 for a new namespace after the original namespace using the value has been deleted.

5 Admin Command Set

...

5.16.1 Log Specific Information

...

Figure 202: Get Log Page – Log Page Identifiers

Log Identifier	Scope	Log Page Name	Reference Section
...
16h	Endurance Group	Rotational Media Information	5.16.1.22
17h	Namespace	Dispersed Namespace Participating NVM Subsystems	5.16.1.NEW
17h 18h to 6Fh	Reserved		
70h		Discovery	5.16.1.23
...
<p>KEY:</p> <p>Namespace = The log page contains information about a specific namespace. Endurance Group = The log page contains information about a specific Endurance Group. Controller = The log page contains information about the controller that is processing the command. Domain = The log page contains information about the Domain. NVM subsystem = The log page contains information about the NVM subsystem. Vendor Specific = The log page contains information that is vendor specific.</p>			
<p>NOTES:</p> <ol style="list-style-type: none"> For namespace identifiers of 0h or FFFFFFFFh. For namespace identifiers other than 0h or FFFFFFFFh. Bit 0 is cleared to '0' in the DSTO field in the Identify Controller data structure (refer to Figure 275). Bit 0 is set to '1' in the DSTO field in the Identify Controller data structure. Selection of a UUID may be supported. Refer to section 8.25. For NVM subsystems that support multiple domains (refer to the MDS bit in the Identify Controller data structure, Figure 275), Domain scope information is returned. 			

5.16.1.NEW Dispersed Namespace Participating NVM Subsystems (Log Identifier 17h)

This log page is used to provide a list of NQNs for all participating NVM subsystems which contain controllers that are able to provide access to the dispersed namespace for the specified NSID. If an NSID of 0h or FFFFFFFFh is specified by the host in the Get Log Page command, then the controller shall abort that command with a status code of Invalid Namespace or Format as described in Figure 87. If the NSID specified by the host in the Get Log Page command is associated with a namespace that is not a dispersed namespace (i.e., the Dispersed Namespace (DISNS) bit is cleared to '0' in the Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC) field of the Identify Namespace data structure associated with that namespace), then the controller shall abort that command with a status code of Invalid Field in Command as described in section 5.16. The method used by the NVM subsystem containing the controller processing the command to discover the NQNs of separate participating NVM subsystems which contain controllers that are able to provide access to the same dispersed namespace is outside the scope of this specification. Each entry in the list shall contain an NQN that matches the NQN contained in the NVM Subsystem NVMe Qualified Name (SUBNQN) field of that participating NVM subsystem's Identify Controller data structure (refer to Figure 275). The first entry in the list shall always contain the NQN of the NVM subsystem containing the controller processing the command.

The log page returned is defined in Figure **NEW.1**.

Figure NEW.1: Get Log Page – Dispersed Namespace Participating NVM Subsystems Log

Bytes	Description
07:00	Generation Counter (GENCTR): This field contains a value that is incremented each time the participating NVM subsystem information for the specified NSID changes (i.e., each time participating NVM subsystems are added or removed for the specified NSID since the last time the host read this log page. 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 of Participating NVM Subsystems (NUMPSUB): This field indicates the number of participating NVM subsystems contained in the log.
255:16	Reserved
511:256	Participating NVM Subsystem Entry 0 (PSUBE0): This field contains the NQN of the NVM subsystem containing the controller processing the command.
767:512	Participating NVM Subsystem Entry 1 (PSUBE1): This field contains the NQN of a participating NVM subsystem that contains controllers that are able to provide access to the dispersed namespace.
...	...
$((N + 2) * 256) - 1$: $((N + 1) * 256)$	Participating NVM Subsystem Entry N (PSUBEN): This field contains the NQN of a participating NVM subsystem that contains controllers that are able to provide access to the dispersed namespace.

...

5.17.2.8 I/O Command Set Independent Identify Namespace data structure (CNS 08h)

...

Figure 280: Identify – I/O Command Set Independent Identify Namespace Data Structure

Bytes	O/M ¹	Description								
00	M	<p>Common Namespace Features (NSFEAT): This field defines features of the namespace.</p> <p>Bits 7:5 are reserved.</p> <p>Bit 4 (RMEDIA) if set to '1' indicates that the namespace stores data on rotational media (refer to section 8.20). If cleared to '0', indicates that the namespace does not store data on rotational media.</p> <p>Bit 3 (UIDREUSE) if set to '1' indicates that the value in the NGUID field for this namespace, if non-zero, is never reused by the controller and that the value in the EUI64 field for this namespace, if non-zero, is never reused by the controller. If cleared to '0', then the NGUID value may be reused and the EUI64 value may be reused by the controller for a new namespace created after this namespace is deleted. This bit shall be cleared to '0' if both NGUID and EUI64 fields are cleared to 0h. Refer to section 4.5.1.</p> <p>Bit 2:1 are reserved.</p> <p>Bit 0 (Rotational Media) if set to '1' indicates that the namespace is associated with rotational media (refer to section 8.20). If cleared to '0', then the namespace is not associated with rotational media.</p>								
01	O	<p>Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC): This field specifies multi-path I/O and namespace sharing capabilities of the namespace.</p> <table border="1"> <thead> <tr> <th>Bits</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>Bits 7:4:2</td> <td>are Reserved</td> </tr> <tr> <td>1</td> <td>Dispersed Namespace (DISNS): If set to '1', then the namespace is a dispersed namespace (refer to section 8.20). If this bit is set to '1', then the namespace shall be a shared namespace (i.e., the Shared Namespace (SHRNS) bit shall also be set to '1'). If cleared to '0', then the namespace is not a dispersed namespace.</td> </tr> <tr> <td>Bit 0</td> <td>Shared Namespace (SHRNS): If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). If cleared to '0', then the namespace is a private namespace and is able to be attached to only one controller at a time. <Find all references to Bit 0 in the NMIC field throughout the spec and change them to "the Shared Namespace (SHRNS) bit".></td> </tr> </tbody> </table>	Bits	Description	Bits 7:4:2	are Reserved	1	Dispersed Namespace (DISNS): If set to '1', then the namespace is a dispersed namespace (refer to section 8.20). If this bit is set to '1', then the namespace shall be a shared namespace (i.e., the Shared Namespace (SHRNS) bit shall also be set to '1'). If cleared to '0', then the namespace is not a dispersed namespace.	Bit 0	Shared Namespace (SHRNS): If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). If cleared to '0', then the namespace is a private namespace and is able to be attached to only one controller at a time. <Find all references to Bit 0 in the NMIC field throughout the spec and change them to "the Shared Namespace (SHRNS) bit".>
Bits	Description									
Bits 7:4:2	are Reserved									
1	Dispersed Namespace (DISNS): If set to '1', then the namespace is a dispersed namespace (refer to section 8.20). If this bit is set to '1', then the namespace shall be a shared namespace (i.e., the Shared Namespace (SHRNS) bit shall also be set to '1'). If cleared to '0', then the namespace is not a dispersed namespace.									
Bit 0	Shared Namespace (SHRNS): If set to '1', then the namespace may be attached to two or more controllers in the NVM subsystem concurrently (i.e., may be a shared namespace). If cleared to '0', then the namespace is a private namespace and is able to be attached to only one controller at a time. <Find all references to Bit 0 in the NMIC field throughout the spec and change them to "the Shared Namespace (SHRNS) bit".>									

Figure 280: Identify – I/O Command Set Independent Identify Namespace Data Structure

Bytes	O/M ¹	Description
02	O	<p>Reservation Capabilities (RESCAP): This field indicates the reservation capabilities of the namespace. A value of 0h in this field indicates that reservations are not supported by this namespace. Refer to section 8.19 for more details.</p> <p>Bit 7 if set to '1' indicates that Ignore Existing Key is used as defined in NVM Express Base Specification revision 1.3 or later. Bit 7 if cleared to '0' indicates that Ignore Existing Key is used as defined in NVM Express Base Specification revision 1.2.1 or earlier. This bit shall be set to '1' if the controller supports revision 1.3 or later as indicated in the Version register.</p> <p>Bit 6 if set to '1' indicates that the namespace supports the Exclusive Access – All Registrants reservation type. If this bit is cleared to '0', then the namespace does not support the Exclusive Access – All Registrants reservation type.</p> <p>Bit 5 if set to '1' indicates that the namespace supports the Write Exclusive – All Registrants reservation type. If this bit is cleared to '0', then the namespace does not support the Write Exclusive – All Registrants reservation type.</p> <p>Bit 4 if set to '1' indicates that the namespace supports the Exclusive Access – Registrants Only reservation type. If this bit is cleared to '0', then the namespace does not support the Exclusive Access – Registrants Only reservation type.</p> <p>Bit 3 if set to '1' indicates that the namespace supports the Write Exclusive – Registrants Only reservation type. If this bit is cleared to '0', then the namespace does not support the Write Exclusive – Registrants Only reservation type.</p> <p>Bit 2 if set to '1' indicates that the namespace supports the Exclusive Access reservation type. If this bit is cleared to '0', then the namespace does not support the Exclusive Access reservation type.</p> <p>Bit 1 if set to '1' indicates that the namespace supports the Write Exclusive reservation type. If this bit is cleared to '0', then the namespace does not support the Write Exclusive reservation type.</p> <p>Bit 0 if set to '1' indicates that the namespace supports the Persist Through Power Loss capability. If this bit is cleared to '0', then the namespace does not support the Persist Through Power Loss Capability.</p>
...
4095:15		Reserved
NOTES:		
1. O/M definition: O = Optional, M = Mandatory.		

...

5.27.1.18 Host Behavior Support (Feature Identifier 16h)

...

Figure 351: Host Behavior Support – Data Structure

Bytes	Description
...	...
02	<p>LBA Format Extension Enable (LBAFEE): I/O Command Set specific definition. Refer to the applicable I/O Command Set specification for details.</p> <p>All values other than 0h and 1h are reserved.</p>

Figure 351: Host Behavior Support – Data Structure

Bytes	Description
03	<p>Host Dispersed Namespace Support (HDISNS): If set to 1h, then dispersed namespaces are enabled. A host that sets this field to 1h specifies that globally unique namespace identifiers (refer to section 8.NEW.2) are used for identifying namespaces (e.g., multi-path I/O software does not use the NSID of a namespace as the sole method for detection of multiple paths to that namespace).</p> <p>A host that supports reservations (refer to section 8.19) and sets this field to 1h specifies a Host Identifier that is unique across all hosts that connect to any participating NVM subsystem.</p> <p>If cleared to 0h, then dispersed namespaces are disabled. If the participating NVM subsystem prohibits host access to dispersed namespaces (refer to section 8.NEW.3.1) when this field is cleared to 0h, then the controller aborts any I/O commands (refer to section 7 in this specification and the I/O Commands section in the appropriate I/O Command Set specification) or any of the Admin commands listed in Figure NEW.7 that the host submits to dispersed namespaces with a status code of Host Dispersed Namespace Support Not Enabled, as described in section 8.NEW.3.1.</p> <p>All values other than 0h and 1h are reserved.</p>
511:0304	Reserved

6 Fabrics Command Set

...

6.3 Connect Command and Response

...

The NVM subsystem shall not allocate a Controller ID in the range FFF0h to FFFFh as a valid Controller ID on completion of a Connect command. Controller IDs FFF0h to FFFFh are defined in Figure NEW.A. If the host is not allowed to establish an association to any controller in the NVM subsystem, then a status code of Connect Invalid Host is returned.

...

A status code of Connect Invalid Parameters is returned for a Connect command if:

- the host sends a Connect command to create an I/O Queue while the controller is disabled;
- the Host Identifier, Host NQN, NVM Subsystem NQN, and the Controller ID values specified for an I/O Queue are not the same as the values specified for the associated Admin Queue in which the association between the host and controller was established;
- the Host NQN or NVM Subsystem NQN values do not match the values that the NVM subsystem is configured to support;
- there is a syntax error in the Host NQN or NVM Subsystem NQN value (refer to section 4.4); ~~or~~
- the Host Identifier is cleared to 0h; ~~or~~
- the host specifies a Controller ID value in the range FFF0h to FFFDh.

...

7 I/O Commands

...

7.2 Reservation Acquire command

...

Figure 392: Reservation Acquire – Command Dword 10

Bits	Description										
31:16	Reserved										
15:08	Reservation Type (RTYPE): This field specifies the type of reservation to be created. The field is defined in Figure 394.										
07:0405	Reserved										
04	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>										
03	Ignore Existing Key (IEKEY): If this bit is set to a '1', the controller shall return an error of Invalid Field In Command. If this bit is cleared to '0', then the Current Reservation Key is checked.										
02:00	<p>Reservation Acquire Action (RACQA): This field specifies the action that is performed by the command.</p> <table border="1"> <thead> <tr> <th>RACQA Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Acquire</td> </tr> <tr> <td>001b</td> <td>Preempt</td> </tr> <tr> <td>010b</td> <td>Preempt and Abort</td> </tr> <tr> <td>011b to 111b</td> <td>Reserved</td> </tr> </tbody> </table>	RACQA Value	Description	000b	Acquire	001b	Preempt	010b	Preempt and Abort	011b to 111b	Reserved
RACQA Value	Description										
000b	Acquire										
001b	Preempt										
010b	Preempt and Abort										
011b to 111b	Reserved										

...

7.3 Reservation Register command

...

Figure 396: Reservation Register – Command Dword 10

Bits	Description										
31:30	<p>Change Persist Through Power Loss State (CPTPL): This field allows the Persist Through Power Loss (PTPL) state associated with the namespace to be modified as a side effect of processing this command. If the Reservation Persistence Feature (refer to section 5.27.1.27) is saveable, then any change to the PTPL state as a result of processing this command shall be applied to both the current value and the saved value of that feature.</p> <table border="1"> <thead> <tr> <th>CPTPL Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>00b</td> <td>No change to PTPL state</td> </tr> <tr> <td>01b</td> <td>Reserved</td> </tr> <tr> <td>10b</td> <td>Clear PTPL state to '0'. Reservations are released and registrants are cleared on a power on.</td> </tr> <tr> <td>11b</td> <td>Set PTPL state to '1'. Reservations and registrants persist across a power loss.</td> </tr> </tbody> </table>	CPTPL Value	Description	00b	No change to PTPL state	01b	Reserved	10b	Clear PTPL state to '0'. Reservations are released and registrants are cleared on a power on.	11b	Set PTPL state to '1'. Reservations and registrants persist across a power loss.
CPTPL Value	Description										
00b	No change to PTPL state										
01b	Reserved										
10b	Clear PTPL state to '0'. Reservations are released and registrants are cleared on a power on.										
11b	Set PTPL state to '1'. Reservations and registrants persist across a power loss.										
29:0405	Reserved										

Figure 396: Reservation Register – Command Dword 10

Bits	Description										
04	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>										
03	<p>Ignore Existing Key (IEKEY): If this bit is set to a '1', then Reservation Register Action (RREGA) field values that use the Current Reservation Key (CRKEY) shall succeed regardless of the value of the Current Reservation Key field in the command (i.e., the current reservation key is not checked).</p>										
02:00	<p>Reservation Register Action (RREGA): This field specifies the registration action that is performed by the command.</p> <table border="1"> <thead> <tr> <th>RREGA Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Register Reservation Key</td> </tr> <tr> <td>001b</td> <td>Unregister Reservation Key</td> </tr> <tr> <td>010b</td> <td>Replace Reservation Key</td> </tr> <tr> <td>011b to 111b</td> <td>Reserved</td> </tr> </tbody> </table>	RREGA Value	Description	000b	Register Reservation Key	001b	Unregister Reservation Key	010b	Replace Reservation Key	011b to 111b	Reserved
RREGA Value	Description										
000b	Register Reservation Key										
001b	Unregister Reservation Key										
010b	Replace Reservation Key										
011b to 111b	Reserved										

...

7.4 Reservation Release command

...

Figure 399: Reservation Release – Command Dword 10

Bits	Description
31:16	Reserved
15:08	<p>Reservation Type (RTYPE): If the Reservation Release Action field is cleared to 000b (i.e., Release), then this field specifies the type of reservation that is being released. The reservation type in this field shall match the current reservation type. If the reservation type in this field does not match the current reservation type, then the controller should return a status code of Invalid Field In Command. This field is defined in Figure 394.</p>
07:0405	Reserved

Figure 399: Reservation Release – Command Dword 10

Bits	Description								
04	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>								
03	<p>Ignore Existing Key (IEKEY): If this bit is set to a '1', the controller shall return an error of Invalid Field in Command. If this bit is cleared to '0', then the Current Reservation Key is checked.</p>								
02:00	<p>Reservation Release Action (RRELA): This field specifies the reservation action that is performed by the command.</p> <table border="1"> <thead> <tr> <th>Value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>000b</td> <td>Release</td> </tr> <tr> <td>001b</td> <td>Clear</td> </tr> <tr> <td>010b to 111b</td> <td>Reserved</td> </tr> </tbody> </table>	Value	Description	000b	Release	001b	Clear	010b to 111b	Reserved
Value	Description								
000b	Release								
001b	Clear								
010b to 111b	Reserved								

...

7.5 Reservation Report command

The Reservation Report command returns a Reservation Status data structure to memory that describes the registration and reservation status of a namespace.

If the namespace is not a dispersed namespace, then the size of the Reservation Status data structure is a function of the number of controllers in the NVM subsystem that are associated with hosts that are registrants of the namespace (i.e., there is a Registered Controller data structure and/or Registered Controller extended data structure for each such controller). If the namespace is a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems, then the size of the Reservation Status data structure is a function of the number of controllers in the NVM subsystem containing the controller processing the command that are associated with hosts that are registrants of the namespace and the number of hosts that are registrants of the namespace in each separate participating NVM subsystem. The controller returns the data structure in Figure 404 if the host has selected a 64-bit Host Identifier and the data structure in Figure 405 if the host has selected a 128-bit Host Identifier (refer to section 5.27.1.25).

...

Figure 403: Reservation Report – Command Dword 11

Bits	Description
31:0402	Reserved

Figure 403: Reservation Report – Command Dword 11

Bits	Description
01	<p>Dispersed Namespace Reservation Support (DISNSRS): This bit specifies host support for reservations on dispersed namespaces for a host that supports dispersed namespaces (i.e., for a host that has set the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature).</p> <p>If this bit is set to '1', then the host supports reservations on dispersed namespaces (i.e., the host supports receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)).</p> <p>If this bit is cleared to '0', then the host does not support reservations on dispersed namespaces (i.e., the host does not support receiving a value of FFFDh in the Controller ID (CNTLID) field of a Registered Controller data structure or Registered Controller Extended data structure (refer to section 8.NEW.5)). If the HDISNS field is set to 1h in the Host Behavior Support feature and the host submits the command to a dispersed namespace with this bit cleared to '0', then the controller aborts the command with a status code of Namespace Is Dispersed as described in section 8.NEW.5.</p>
00	<p>Extended Data Structure (EDS): If set to '1', then the controller returns the extended data structure defined in Figure 405. If cleared to '0', then the controller returns the data structure defined in Figure 404.</p>

...

Figure 406: Registered Controller Data Structure

Bytes	Description
01:00	<p>Controller ID (CNTLID): This field contains the controller ID (i.e., the value of the CNTLID field in the Identify Controller data structure) of the controller whose status is reported in this data structure.</p> <p>If the controller is a dynamic controller (refer to section 3.1.1) that is not associated with a host, then the Controller ID field shall be set to FFFFh.</p> <p>If the namespace is a dispersed namespace and the controller is not contained in the same participating NVM subsystem as the controller processing the command, then the Controller ID field is set to FFFDh, as described in section 8.NEW.5.</p>
02	<p>Reservation Status (RCSTS): This field indicates the reservation status of the controller described by this data structure.</p> <p>Bits 7:1 are reserved</p> <p>Bit 0 is set to '1' if the controller is associated with a host that holds a reservation on the namespace.</p>
07:03	Reserved
15:08	<p>Host Identifier (HOSTID): This field contains the 64-bit Host Identifier of the controller described by this data structure.</p>
23:16	<p>Reservation Key (RKEY): This field contains the reservation key of the host associated with the controller described by this data structure.</p>

...

8 Extended Capabilities

...

8.1.2 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 (refer to section 2.3.1). The method for assigning namespaces to ANA Groups is outside the scope of this 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 275).

The ANA Group Identifier (ANAGRPID) for each ANA Group shall be unique within the NVM subsystem. If bit 6 in the ANACAP field in the Identify Controller data structure is set to '1', then the ANA Group Identifier shall not change while the namespace is attached to any controller in the NVM subsystem. If bit 6 in the ANACAP field is cleared to '0', then the ANA Group Identifier may change while the namespace is attached to any controller in the NVM subsystem. If the ANA Group Identifier changes, the controller shall issue the Asymmetric Namespace Access Change Notice as described in 8.1.3.6.

If two or more participating NVM subsystems provide access to a dispersed namespace, and that dispersed namespace is a member of an ANA Group on any participating NVM subsystem, then each participating NVM subsystem shall contain an ANA Group whose members include that dispersed namespace. The ANAGRPID for a dispersed namespace may or may not be the same on each participating NVM subsystem.

<ANA grouping requirements about each participating NVM subsystem maintaining the same ANA Group membership for overlapping sets of dispersed namespaces has been removed. For example, it's now permissible to have ANA Group 1 on NVM Subsystem 1 contain dispersed namespaces A and B together and for ANA Group 2 on NVM Subsystem 2 to contain only dispersed namespace A and for ANA Group 3 on NVM Subsystem 2 to contain only dispersed namespace B.>

...

...

8.10.1 Host ANA Normal Operation

The host determines if ANA is supported by examining bit 3 in the CMIC field in the Identify Controller data structure (refer to Figure 275). The NSID or Identifier (refer to section 4.3) is used to determine when multiple paths to the same namespace ~~that is not a dispersed namespace~~ are available. ~~For dispersed namespaces, globally unique namespace identifiers (refer to section 8.NEW.2) are used to determine when multiple paths are available to the same namespace, as described in section 8.NEW.4.~~ The host examines the ANA log page (refer to section 5.16.1.13) for each controller to determine the ANA state of each group of namespaces attached to that controller.

...

...

8.19 Reservations

...

Controllers that make up an NVM subsystem shall all have the same support for reservations. Although strongly encouraged, namespaces that make up an NVM subsystem are not all required to have the same support for reservations. For example, some namespaces within a single controller may support reservations while others do not, or the supported reservation types may differ among namespaces. If a controller supports reservations, then the controller shall:

- Indicate support for reservations by returning a '1' in bit 5 of the Optional NVM Command Support (ONCS) field in the Identify Controller data structure;

- Support the Reservation Report command (refer to section 7.5), Reservation Register command (refer to section 7.3), Reservation Acquire command (refer to section 7.1), and Reservation Release command (refer to section 7.4);
- Support the Reservation Notification log page;
- Support the Reservation Log Page Available asynchronous events;
- Support the Reservation Notification Mask Feature;
- Support the Host Identifier Feature; and
- Support the Reservation Persistence Feature.

If a controller supports dispersed namespaces and supports reservations, then the controller supports the Dispersed Namespace Reservation Support (DISNSRS) bit being set to '1' in the Reservation Report command, Reservation Register command, Reservation Acquire command, and Reservation Release command, as described in section 8.NEW.5.

If a namespace supports reservations, then the namespace shall:

- Report a non-zero value in the Reservation Capabilities (RESCAP) field in the Identify Namespace data structure;
- Support Persist Through Power Loss (PTPL) state; and
- Support sufficient resources to allow a host to successfully register a reservation key on every controller in the NVM subsystem with access to the shared namespace (i.e., a Reservation Register command shall never fail due to lack of resources).

...

...

8.19.3 Registering

...

Registering a reservation key with a namespace creates an association between a host and a namespace. A host that is a registrant of a namespace may use any controller with which that host is associated (i.e., that has the same Host Identifier, refer to section 5.27.1.25) to access that namespace as a registrant. Thus, a host is only required to register on a single controller to become a registrant of the namespace on all controllers in the NVM subsystem that have access to the namespace and are associated with the host. If a host attempts to access a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems, then that host is only required to register on a single controller in a single participating NVM subsystem to become a registrant of the dispersed namespace on all controllers in all participating NVM subsystems that have access to the dispersed namespace and are associated with the host.

...

A host that is a registrant of a namespace may replace the existing reservation key value for that namespace by executing a Reservation Register command on the namespace with the:

- a) RREGA field set to 010b (i.e., Replace Reservation Key);
- b) current reservation key in the Current Reservation Key (CRKEY) field; and
- c) new reservation key in the NRKEY field.

The current reservation key value shall be replaced by the new reservation key value in all controllers to which the namespace is attached that have the same Host Identifier as the Host Identifier of the controller processing the command. For dispersed namespaces, this requirement includes all participating NVM subsystems. If the contents of the CRKEY field do not match the key currently associated with the host, then the command shall be aborted with a status code of Reservation Conflict. A host may replace its reservation key without regard to its registration status or current reservation key value by setting the Ignore Existing Key (IEKEY) bit to '1' in the Reservation Register command. Replacing a reservation key has no effect on any reservation that may be held on the namespace.

8. NEW Dispersed Namespaces

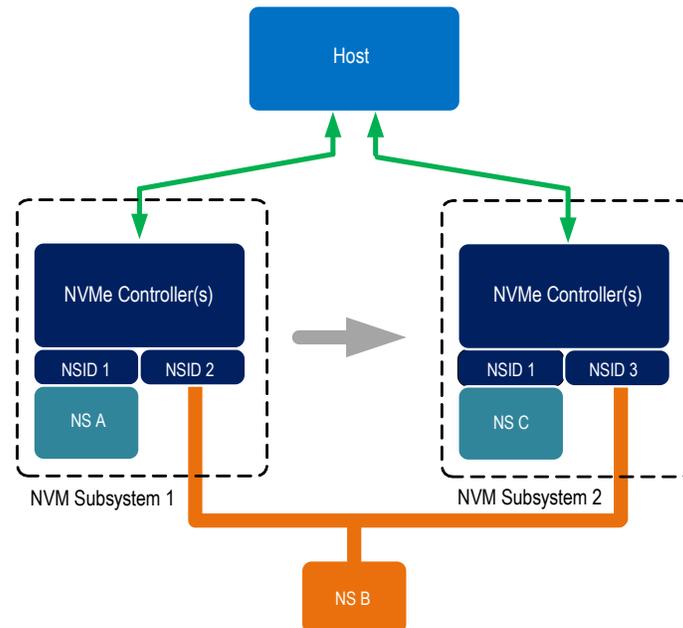
<This will be the model clause.>

Dispersed namespaces provide hosts with access to the same namespace using multiple participating NVM subsystems. Two prominent scenarios that require namespace access from multiple participating NVM subsystems are:

1. online data migration; and
2. data replication.

Online data migration is used to transparently move the contents of one or more namespaces across participating NVM subsystems without disrupting host access to those namespaces during the migration. For each namespace whose contents are migrated, one or more paths to the namespace in the destination NVM subsystem are added to the multi-path I/O on the host, in addition to the existing paths to the namespace in the source NVM subsystem. Once the online data migration completes (i.e., all of the contents of all namespaces being migrated has been successfully migrated from the source NVM subsystem to the destination NVM subsystem), all of the paths to the namespace in the source NVM subsystem are subsequently removed from the multi-path I/O on the host. During the online data migration, the host and participating NVM subsystems must manage the paths so that there is no interruption to I/O. Figure NEW.3 shows an example of online data migration, where the contents of namespace B are migrated from one participating NVM subsystem to another participating NVM subsystem. In this figure, the lines from the host to the NVMe controllers may represent multiple connections from the host to multiple controllers in each participating NVM subsystem.

Figure NEW.3: Online Data Migration



Data replication is used to replicate the contents of one or more dispersed namespaces across participating NVM subsystems such that each participating NVM subsystem is able to provide host access to the same namespace data. Other behavior (e.g., persistent reservations) must also be maintained between hosts and each participating NVM subsystem. Figure NEW.4, Figure NEW.5, and Figure NEW.6 show different examples of data replication, where the contents of namespace B are replicated across two participating NVM subsystems. In these figures, the lines from hosts to NVMe controllers may represent multiple

connections from the hosts to multiple controllers in each participating NVM subsystem. Figure NEW.4 shows an example of single NVM subsystem host connectivity, where two hosts access the same dispersed namespace being replicated across two participating NVM subsystems via controllers in only one NVM subsystem.

Figure NEW.4: Data Replication Example 1

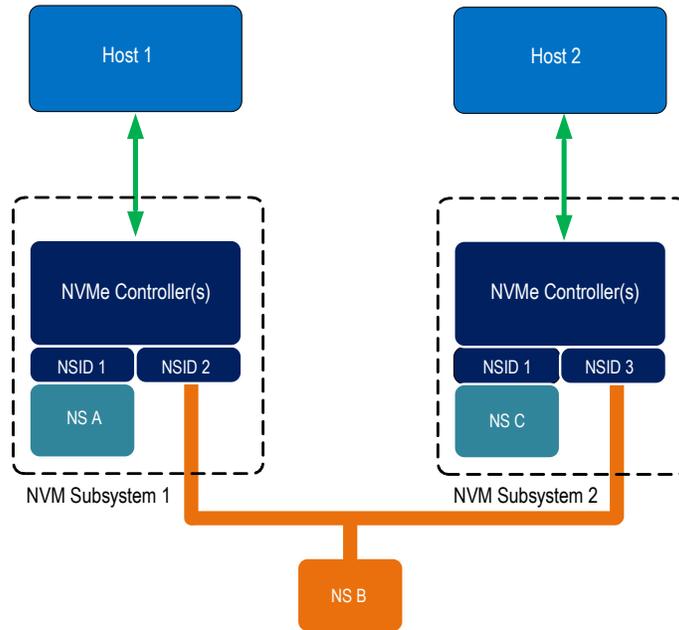


Figure NEW.5 shows another example of single NVM subsystem host connectivity, where two hosts access the same dispersed namespace being replicated across two participating NVM subsystems via controllers in only one NVM subsystem, but the host connected to NVM Subsystem 2 only accesses namespace B in the event of a failure scenario.

Figure NEW.5: Data Replication Example 2

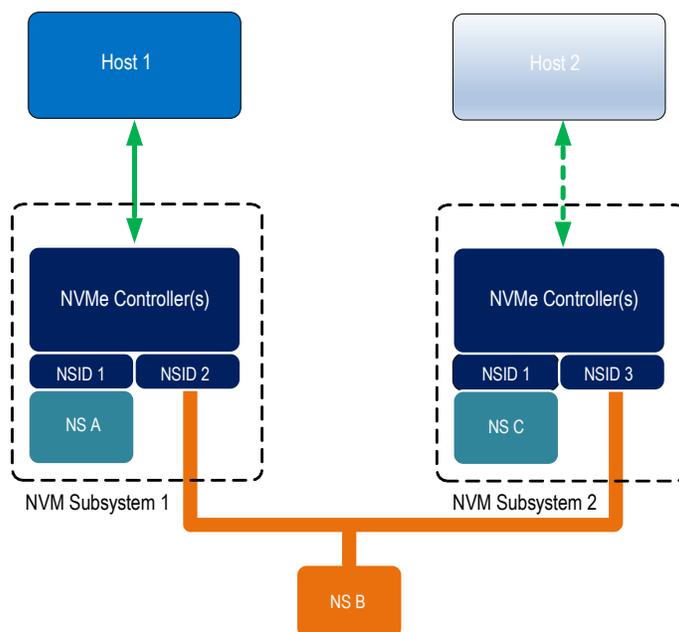
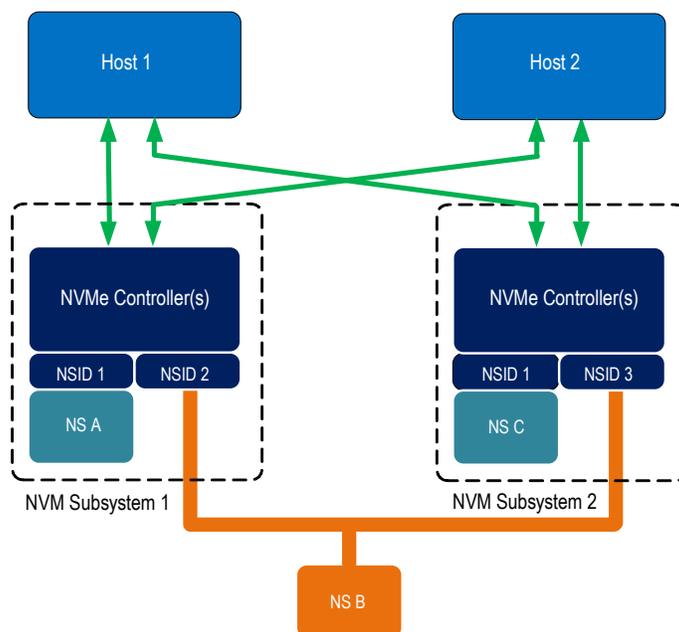


Figure NEW.6 shows an example of multiple NVM subsystem host connectivity, where two hosts access the same dispersed namespace being replicated across two participating NVM subsystem via controllers in both NVM subsystems.

Figure NEW.6: Data Replication Example 3



8.NEW.1 Dispersed Namespace Management

The Namespace Management command (refer to section 5.23) may be used to create a namespace that is capable of being accessed using controllers contained in two or more participating NVM subsystems concurrently (i.e., may be used to create a namespace that is capable of being a dispersed namespace). The methods for dispersing a namespace (e.g., attaching that namespace to controllers contained in separate participating NVM subsystems) are outside the scope of this specification. The Namespace Management or Capacity Management command may be used to delete a dispersed namespace on the participating NVM subsystem containing the controller processing the command. Deleting a dispersed namespace on one participating NVM subsystem may or may not affect that namespace on other participating NVM subsystems (e.g., a deletion from one participating NVM subsystem may have no effect on the namespace on other participating NVM subsystems, or a deletion from one participating NVM subsystem may delete the namespace from all participating NVM subsystems).

The Namespace Management command may be used to create a shared namespace that is not a dispersed namespace which is later converted to a dispersed namespace. The method of converting a namespace that is not a dispersed namespace to a dispersed namespace is outside the scope of this specification. NVM subsystems should not convert private namespaces to dispersed namespaces. Whenever a namespace that is not a dispersed namespace is converted to a dispersed namespace or a dispersed namespace is converted to a namespace that is not a dispersed namespace, the controller reports a Namespace Attribute Changed event as described in Figure 146.

The host may attach or detach a dispersed namespace from the controller by using the Namespace Attachment command (refer to section 5.22).

8.NEW.2 NSID and Globally Unique Namespace Identifier Usage

The NSID for a dispersed namespace is unique for all controllers in a participating NVM subsystem, as described in section 3.2.1.6. The NSID for a dispersed namespace may be different on different participating NVM subsystems.

A dispersed namespace has a globally unique namespace identifier (refer to section 4.5.1.4) that uniquely identifies the namespace in all participating NVM subsystems. The globally unique namespace identifier has the same value for that namespace in all participating NVM subsystems and shall be used to determine which NSIDs on different participating NVM subsystems refer to the same namespace. The globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems shall be either:

- a) a Namespace Globally Unique Identifier (NGUID); or
- b) a Universally Unique Identifier (UUID).

If the dispersed namespace has an NGUID and does not have a UUID, then:

- a) that NGUID shall be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems;
- b) all participating NVM subsystems shall use the same NGUID value for the dispersed namespace; and
- c) the dispersed namespace may have an EUI64 that shall not be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems.

If the dispersed namespace has a UUID and does not have an NGUID, then:

- a) that UUID shall be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems;
- b) all participating NVM subsystems shall use the same UUID value for the dispersed namespace; and
- c) the dispersed namespace may have an EUI64 that shall not be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems.

If the dispersed namespace has an NGUID and has a UUID, then:

- a) the NGUID shall be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems;
- b) all participating NVM subsystems shall use the same NGUID value for the dispersed namespace; and
- c) the dispersed namespace may have an EUI64 that shall not be the globally unique namespace identifier that uniquely identifies the dispersed namespace in all participating NVM subsystems.

8.NEW.3 Dispersed Namespace Access

The host may discover if a namespace is capable of being accessed using controllers contained in two or more participating NVM subsystems concurrently (i.e., may discover if a namespace may be a dispersed namespace) by issuing an Identify command (i.e., CNS 00h if supported by the I/O Command Set or CNS 08h) to the controller for the specified NSID. If the namespace is capable of being accessed using controllers contained in two or more participating NVM subsystems concurrently, then the controller shall set the Dispersed Namespace (DISNS) bit to '1' in the Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC) field of the returned I/O Command Set Independent Identify Namespace data structure (refer to Figure 280) or I/O Command Set specific Identify Namespace data structure (refer to the NVM Command Set Specification).

All controllers in each participating NVM subsystem that are able to provide access to a specific dispersed namespace shall support the I/O Command Set associated with that dispersed namespace.

The host may discover the NQNs of participating NVM subsystems which contain controllers that are able to provide access to a dispersed namespace by issuing a Get Log Page command with the Log Page Identifier (LID) field set to 17h (refer to section 5.16.1.NEW).

Hosts specify support for dispersed namespaces by setting the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature by issuing a Set Features command with Feature Identifier (FID) set to 16h (refer to Figure 351).

If the HDISNS field is set to 1h, the host specifies support for:

- treating all instances of a dispersed namespace as the same namespace, based upon the shared globally unique namespace identifier (refer to section 8.NEW.2) when accessing that dispersed namespace through multiple controllers on the same participating NVM subsystem;
- treating all instances of a dispersed namespace as the same namespace, based upon the shared globally unique namespace identifier when accessing that dispersed namespace through controllers on multiple participating NVM subsystems; and
- specifying a Host Identifier that is unique across all hosts that connect to any participating NVM subsystem.

If the HDISNS field is set to 1h, the controller shall not abort any command that the host submits to dispersed namespaces with a status code of Host Dispersed Namespace Support Not Enabled.

<There could be corner cases where an old host (HDISNS = 0) will get RESERVATION CONFLICT but not be able to find out why if it can't get back CNTLID = FFFDh. After consulting with a number of host developers, it was determined that getting back FFFDh will not be a problem, so the requirement that HDISNS = 1 before controllers can report CNTLID = FFFDh has been removed.>

A participating NVM subsystem shall support prohibiting host access to dispersed namespaces when the HDISNS field is cleared to 0h in the Host Behavior Support feature, as described in section 8.NEW.3.1. A participating NVM subsystem may either:

- allow host access to a dispersed namespace when the HDISNS field is cleared to 0h in the Host Behavior Support feature if the host associated with the Host NQN in the connection is only able to access the dispersed namespace from a single participating NVM subsystem; or
- prohibit host access to dispersed namespaces when the HDISNS field is cleared to 0h in the Host Behavior Support feature.

A participating NVM subsystem that allows host access to a dispersed namespace when the HDISNS field is cleared to 0h in the Host Behavior Support feature if the host associated with the Host NQN in the connection is only able to access the dispersed namespace from a single participating NVM subsystem shall implement a method to determine if the host is able to access the dispersed namespace from multiple participating NVM subsystems. The method used to determine if the host is able to access the dispersed namespace from multiple participating NVM subsystems is outside the scope of this specification. If a participating NVM subsystem determines that a Connect command requests a host with the HDISNS field cleared to 0h in the Host Behavior Support feature having access to the same dispersed namespace through multiple participating NVM subsystems, then that participating NVM subsystem shall either:

- a) abort a Connect command with a status code of Connect Invalid Host if that Connect command requests the host associated with the Host NQN in the connection having access to a dispersed namespace through multiple participating NVM subsystems. Aborting a Connect command for this reason may result in the host being unable to access other namespaces (i.e., namespaces that are unrelated to the dispersed namespace) on that NVM subsystem; or
- b) allow a Connect command that requests the host associated with the Host NQN in the connection having access to a dispersed namespace through multiple participating NVM subsystems, but prohibit host access to that dispersed namespace from multiple participating NVM subsystems (e.g., prohibit host access to that dispersed namespace on the second participating NVM subsystem) as described in section 8.NEW.3.1.

Option a) in the preceding list is intended to enforce the requirement that a host that has not set the HDISNS field to 1h in the Host Behavior Support feature be allowed to access a dispersed namespace on only one participating NVM subsystem by preventing all access to any other participating NVM subsystems (including all other namespaces on those other participating NVM subsystems). Option b) in the preceding list is intended to enforce the requirement that a host that has not set the HDISNS field to 1h in the Host Behavior Support feature be allowed to access a dispersed namespace on only one participating NVM subsystem by, for each dispersed namespace, preventing access to that dispersed namespace on any other participating NVM subsystems (and allowing access to other namespaces that are not dispersed namespaces on those other participating NVM subsystems).

8.NEW.3.1 Prohibiting Host Access to Dispersed Namespaces

If a participating NVM subsystem prohibits host access to dispersed namespaces when the Host Dispersed Namespace Support (HDISNS) field is cleared to 0h in the Host Behavior Support feature, then the controller shall abort the following commands with a status code of Host Dispersed Namespace Support Not Enabled:

- any I/O commands (refer to section 7 in this specification and the I/O Commands section in the appropriate I/O Command Set specification) that specify the NSID of a dispersed namespace in the command; or
- any of the Admin commands listed in Figure NEW.7 that specify the NSID of a dispersed namespace in the command.

The Additional Restrictions column in Figure NEW.7 lists additional restrictions that apply while the HDISNS field is cleared to 0h in the Host Behavior Support feature for Admin commands that are capable of affecting dispersed namespaces without specifying the NSID of a dispersed namespace.

Figure NEW.7: Dispersed Namespaces Command Restrictions – Prohibited Admin Commands

Admin Command	Additional Restrictions
Directive Receive	None.
Directive Send	None.
Set Features	This command is prohibited if a dispersed namespace is attached to the controller and an NSID of FFFFFFFFh is specified for a feature capable of affecting all namespaces attached to that controller.
Format NVM	This command is prohibited if a dispersed namespace is allocated in the participating NVM subsystem and the scope of the command includes all allocated namespaces in that NVM subsystem (refer to Figure 188). This command is prohibited if a dispersed namespace is attached to the controller and the scope of the command includes all namespaces attached to that controller (refer to Figure 188).

8.NEW.4 ANA Considerations

If more than one participating NVM subsystem contains controllers that provide a host with access to a dispersed namespace, then Asymmetric Namespace Access (ANA) and Asymmetric Namespace Access Reporting (refer to section 8.1) should be supported in all participating NVM subsystems. If ANA is used with dispersed namespaces, then globally unique namespace identifiers (refer to section 8.NEW.2) are used to determine when multiple paths are available to the same dispersed namespace. Hosts specify to controllers that their host software (e.g., multipath I/O software) uses globally unique namespace identifiers to determine when multiple paths are available to the dispersed namespace by setting the HDISNS field to 1h in the Host Behavior Support feature, as described in section 8.NEW.3.

ANA Group usage with dispersed namespaces is described in section 8.1.2.

8.NEW.5 Reservation Considerations

Reservations for a dispersed namespace that is able to be accessed by controllers in multiple participating NVM subsystems are intended to be coordinated between each participating NVM subsystem. To ensure the uniqueness of a host's identity and prevent potential data corruption, each host that sets the Host Dispersed Namespace Support (HDISNS) field to 1h in the Host Behavior Support feature specifies a Host Identifier that is unique across all hosts that connect to any participating NVM subsystem, as described in section 8.NEW.3.

If a host is a reservation registrant on a dispersed namespace, and that host submits a Reservation Report command to that namespace, then for reservations established using controllers that are not contained in the same participating NVM subsystem as the controller processing the Reservation Report command, the Controller ID (CNTLID) field shall be set to FFFDh in the Registered Controller data structure (refer to Figure 406) or Registered Controller Extended data structure (refer to Figure 407). Multiple identical Registered Controller or Registered Controller Extended data structures with the CNTLID field set to FFFDh shall not be returned by a single Reservation Report command; as a result, a Registered Controller or Registered Controller Extended data structure with the CNTLID field set to FFFDh indicates one or more controllers are associated with a host that is a registrant of the dispersed namespace in another participating NVM subsystem. A Registered Controller or Registered Controller Extended data structure with the CNTLID field set to FFFDh is returned for each host (i.e., as identified by the HOSTID field in that data structure) that is a registrant of the dispersed namespace in any other participating NVM subsystem.

The Dispersed Namespace Reservation Support (DISNSRS) bit in a command is set to '1' by hosts that support reservations on dispersed namespaces (i.e., hosts that support receiving a value of FFFDh in the CNTLID field of a Registered Controller data structure or Registered Controller Extended data structure). A controller that supports dispersed namespaces and supports reservations (i.e., a controller that has Bit 5 set to '1' in the Optional NVM Command Support (ONCS) field of the Identify Controller data structure (refer to Figure 275)) shall support the DISNSRS bit being set to '1' in each command in the following list:

- Reservation Acquire (refer to section 7.2);
- Reservation Register (refer to section 7.3);
- Reservation Release (refer to section 7.4); and
- Reservation Report (refer to section 7.5).

If the HDISNS field is set to 1h in the Host Behavior Support feature and the DISNSRS bit is cleared to '0' in any of the commands in the preceding list when submitted to a dispersed namespace, then the controller shall abort that command with a status code of Namespace Is Dispersed.

Description of NVM Command Set Specification 1.0 Changes

2 NVM Command Set Model

...

2.1.4 Atomic Operations

...

In the case of a shared namespace (e.g., a dispersed namespace or shared namespace that is not a dispersed namespace), operations performed by an individual controller are atomic to the shared namespace at the write atomicity level reported in the corresponding Identify Controller or Identify Namespace data structures of the controller to which the command was submitted.

...

<5/26: FMDS group decided to remove all new requirements for write atomicity in section 2.1.4. Instead, a new paragraph was added to the end of this section to state why it's better to use the same write atomicity level across all controllers for dispersed namespaces (as a recommendation).>

<7/21: FMDS group decided to get rid of the recommendation paragraph entirely, as there could be a good reason why atomicity might differ between participating NVM subsystems (e.g., for the data migration use case). Now, the only time dispersed namespaces are mentioned with regards to atomicity is seen in the small addition to the existing paragraph in 2.1.4 shown above.>