LEGAL NOTICE:

This erratum to the NVM Express revision 1.3 specification is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS: Members of NVM Express, Inc. have the right to use and implement this erratum to the NVM Express revision 1.3 specification subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.: If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: **"© 2007 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

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

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

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

NVM Express™ Technical Errata

Errata ID	**005**
Revision Date	**05/24/2018**
Affected Spec Ver.	**NVM Express™ 1.3b**
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Fred Knight	NetApp Inc.
Feng (Frank) Yang	Huawei
Tom Friend	Toshiba
Tom Friend	SK Hynix
Mike Allison	SK Hynix
Mike Allison	Intel
David Black	Dell EMC
John Geldman	Toshiba
Judy Brock	Samsung

Errata Overview

Reorganization of IDENTIFY command into subsections that show up in the TOC and bookmarks.

Correct spelling errors, and list formatting.

Clarify MSI vs. MSI-X support requirements (relax a “should” to a “may”).

Clarify the MPTR and NSID field in the command format structure.

Clarify error returns for Create I/O Submission Queue command.

Clarify FWUG field use in the Firmware Image Download command.

Clarify Change Namespace List log (make add/delete explicit, rather than implicit).

Clarify which IDENTIFY command CNS values use the CNTID field and which use the NSID field.

Clarify that SANICAP cleared to ‘0’ means sanitize is not supported.

Clarify the interactions between the Autonomous Power State Transition feature and the Non-Operational Power State Config feature.

Clarify that the Namespace Management capability requires both the Namespace Management command and the Namespace Attachment command.

Clarify that some namespace changes may impact the LBA Range Type feature.

Clarify ASQ/ACQ/BMBBA fields split the 64-bit address into a 52-bit field and a 12 bit field.

Add examples of appropriate error code usage for the Virtualization Management command.

Clarify Format command with NSID set to FFFFFFFFh.

Revision History

Revision Date	Change Description
10/09/2017	Initial creation
10/26/2017	Pull Flush command text from ECN-004
01/04/2018	Incorporate IDENTIFY reorganization
01/23/2018	Edits based on committee call, continue incorporating emails; move FLUSH command clarification to TP
01/25/2018	Changes from committee call. ASCII strings should be case insensitive. UTF-8 case debate continues.
01/30/2018	Edits from Judy for MPTR and others.
02/15/2018	Bring in FNA changes from ECN-004; settled on ASCII being case sensitive.
02/27/2018	Move ongoing items into ECN-006; duplicate “52 most significant bits” wording into ASQ/ACQ field descriptions. Move part of DST change to ECN-006
03/06/2018	Incorporate power related changes from John Geldman
03/14/2018	Move Power Management change to ECN-006, SMART critical bit to a TPAR
03/22/2018	Move RTD3 clarification to ECN-006
05/10/2018	In 5.14.1.3 change “namespaces in this controller” to “namespaces attached to this controller” (found during 30 day member review).
05/24/2018	Correct a few section numbers and Figure numbers so they are based on 1.3b.

Incompatible Changes

IDENTIFY command with CNS=10h now explicitly states the ordering requirement of the returned NSID values. This requirement always existed; it could not have worked without this requirement. Refer to section 5.15.6.

The SANICAP field did not provide an indication when the Sanitize command was not supported (the field was reserved if the Sanitize command was not supported). Rather than reserved, if the Sanitize command is not supported, this field is now required to be cleared to zero.

Description of Specification Changes

Modify a portion of section 1.4.1 (Multi-Path I/O and Namespace Sharing) as shown below:

1.4.1 Multi-Path I/O and Namespace Sharing

…

Figure 4 shows a multi-Function NVM ~~Subsystem~~ subsystem with a single PCI Express port containing two controllers, one controller is associated with PCI Function 0 and the other controller is associated with PCI Function 1.

…

Figure 5 illustrates an NVM ~~Subsystem~~ subsystem with two PCI Express ports, each with an associated controller.

…

Modify a portion of section 1.5 (Conventions) as shown below:

1.5 Conventions

Hardware shall return '0' for all bits and registers that are marked as reserved, and host software shall write all reserved bits and registers with the value of '0'.

Inside the register sections (i.e., section 2 and section 3), the following terms and abbreviations are used:

RO	Read Only
RW	Read Write
R/W	Read Write. The value read may not be the last value written.
RWC	Read/Write '1' to clear
~~**RWS**~~	~~Read/Write '1' to set~~
Impl Spec	Implementation Specific – the controller has the freedom to choose its implementation.
HwInit	The default state is dependent on NVM Express controller and system configuration. The value is initialized at reset, for example by an expansion ROM, or in the case of integrated devices, by a platform BIOS.
Reset	This column indicates the value of the field after a reset.

For some register fields, it is implementation specific as to whether the field is RW, RWC, or RO; this is typically shown as RW/RO or RWC/RO to indicate that if the functionality is not supported that the field is read only.

…

Modify a portion of section 2 (System Bus (PCI Express) Registers) as shown below:

2 System Bus (PCI Express) Registers

…

MSI-X is the recommended interrupt mechanism to use. However, some systems ~~do~~ may not support MSI-X, ~~thus~~ as a result, devices ~~should~~ may choose to support both the MSI Capability and the MSI-X Capability.

Modify a portion of section 3.1.9 (Offset 28h: ASQ) as shown below:

3.1.9 Offset 28h: ASQ – Admin Submission Queue Base Address

This register defines the base memory address of the Admin Submission Queue.

Bit	Type	Reset	Description
63:12	RW	Impl Spec	**Admin Submission Queue Base (ASQB):** ~~Indicates~~ This field specifies the 52 most significant bits of the 64-bit physical address for the Admin Submission Queue. This address shall be memory page aligned (based on the value in CC.MPS). All Admin commands, including creation of I/O Submission Queues and I/O Completions Queues shall be submitted to this queue. For the definition of Submission Queues, refer to section 4.1.
11:00	RO	0h	Reserved

Modify a portion of section 3.1.10 (Offset 30h: ACQ) as shown below:

3.1.10 Offset 30h: ACQ – Admin Completion Queue Base Address

This register defines the base memory address of the Admin Completion Queue.

Bit	Type	Reset	Description
63:12	RW	Impl Spec	**Admin Completion Queue Base (ACQB):** ~~Indicates~~ This field specifies the 52 most significant bits of the 64-bit physical address for the Admin Completion Queue. This address shall be memory page aligned (based on the value in CC.MPS). All completion queue entries for the commands submitted to the Admin Submission Queue shall be posted to this Completion Queue. This queue is always associated with interrupt vector 0. For the definition of Completion Queues, refer to section 4.1.
11:00	RO	0h	Reserved

Modify a portion of section 3.1.14 (Offset 44h: BPRSEL) as shown below:

3.1.14 Offset 44h: BPRSEL – Boot Partition Read Select

…

Bit	Type	Reset	Description
31	RW	0h	**Boot Partition Identifier (BPID):** This field specifies the Boot Partition identifier for the Boot ~~Read~~ Partition read operation.
30	RO	0h	Reserved
29:10	RW	0h	**Boot Partition Read Offset (BPROF):** This field selects the offset into the Boot Partition, in 4KB units, that the controller copies into the Boot Partition Memory Buffer.
09:00	RW	0h	**Boot Partition Read Size (BPRSZ):** This field selects the read size in multiples of 4KB to copy into the Boot Partition Memory Buffer.

Modify a portion of section 3.1.15 (Offset 48h: BPMBL) as shown below:

3.1.15 Offset 48h: BPMBL – Boot Partition Memory Buffer Location (Optional)

This optional register specifies the memory buffer that is used as the destination for data when a Boot Partition is read (refer to section 8.13). If the controller does not support the Boot Partitions feature then this register shall be cleared to 0h.

Bit	Type	Reset	Description
63:12	RW	0h	**Boot Partition Memory Buffer Base Address (BMBBA):** This field ~~S~~specifies the 52 most significant bits of the 64-bit physical address for the Boot Partition Memory Buffer. ~~This address shall be 4KB aligned. Note that this field contains the 52 most significant bits of the 64 bit address.~~
11:00	RO	0h	Reserved

Modify a portion of section 4.2 (Submission Queue Entry - Command Format) as shown below (changes shown here are based on changes published in ECN-004):

4.2 Submission Queue Entry - Command Format

…

Figure 11: Command Format – Admin and NVM Command Set

Bytes	Description
03:00	**Command Dword 0 (CDW0):** This field is common to all commands and is defined in Figure 10.
07:04	**Namespace Identifier (NSID):** This field specifies the namespace that this command applies to. If the namespace identifier is not used for the command, then this field shall be cleared to 0h. The value FFFFFFFFh in this field is a broadcast value (refer to section 6.1), where the scope (e.g., the NVM subsystem, all attached namespaces, or all namespaces in the NVM subsystem) is dependent on the command. Refer to Figure 41, and Figure 42 for Admin commands that support the use of the value FFFFFFFFh in this field. Specifying an inactive namespace identifier (refer to section 6.1.4) in a command that uses the namespace identifier shall cause the controller to abort the command with status Invalid Field in Command, unless otherwise specified. Specifying an invalid namespace identifier (refer to section 6.1.2) in a command that uses the namespace identifier shall cause the controller to abort the command with status Invalid Namespace or Format, unless otherwise specified. If the namespace identifier is used for the command (refer to Figure 41 and Figure 42), the value FFFFFFFFh is not supported for that command, and the host specifies a value of FFFFFFFFh, then the controller should abort the command with status Invalid Field in Command, unless otherwise specified. If the namespace identifier is not used for the command and the host specifies a value from 00000001h to FFFFFFFEh, then the controller should abort the command with status Invalid Field in Command, unless otherwise specified.
15:08	Reserved
23:16	**Metadata Pointer (MPTR):** This field is valid only if the command has metadata that is not interleaved with the logical block data, as specified in the Format NVM command. This is a reserved field in NVMe over Fabrics. If CDW0.PSDT (refer to Figure 10) is set to 00b, then this field shall contain the address of a contiguous physical buffer of metadata and that address shall be Dword aligned (i.e., bits 1:0 cleared to 0b). The controller is not required to check that bits 1:0 are cleared to 00b. The controller may report an error of Invalid Field in Command if bits 1:0 are not cleared to 00b. If the controller does not report an error of Invalid Field in Command, then the controller shall operate as if bits 1:0 are cleared to 00b. If CDW0.PSDT is set to 01b, then this field shall contain the address of a contiguous physical buffer of metadata and ~~shall be byte~~ that address may be aligned on any byte boundary. If CDW0.PSDT is set to 10b, then this field shall contain the address of an SGL segment ~~containing~~ that contains exactly one SGL Descriptor ~~and~~. The address of that SGL segment shall be Qword aligned. The SGL Descriptor contained in that SGL segment is the first SGL Descriptor of the metadata for the command. If the SGL Descriptor contained in that SGL segment is an SGL Data Block descriptor, then ~~it~~ that SGL Data Block Descriptor is the only SGL Descriptor and therefore describes the entire metadata data transfer. Refer to section 4.4.
…	

…

Modify a portion of section 5 (Admin Command Set) as shown below (changes shown here are based on changes published in ECN-004):

5 Admin Command Set

…

Figure 12: Opcodes for Admin Commands

Opcode by Field			Combined Opcode[2]	O/M[1]	Namespace Identifier Used[3]	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[4]				
0b	000 00b	00b	00h	M	No	Delete I/O Submission Queue

Opcode by Field			Combined Opcode[2]	O/M[1]	Namespace Identifier Used[3]	Command
(07)	(06:02)	(01:00)				
Generic Command	Function	Data Transfer[4]				
…						
0b	001 10b	01b	19h	O	Yes~~6~~7	Directive Send
0b	001 10b	10b	1Ah	O	Yes~~6~~7	Directive Receive
…						
	Vendor Specific					
1b	na	NOTE 4	C0h – FFh	O		Vendor specific

NOTES:

1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. A subset of commands uses the Namespace Identifier field (CDW1.NSID). If the Namespace Identifier field is used, then the value FFFFFFFFh is supported in this field unless footnote 6 in this figure indicates that a specific command does not support that value. When this field is not used, the field ~~shall be~~ is cleared to 0h as described in Figure 11.
4. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
5. For NVMe over PCIe implementations, the Keep Alive command is optional. For NVMe over Fabrics implementations, the associated NVMe Transport binding defines whether the Keep Alive command is optional or mandatory.
6. This command does not support the use of the Namespace Identifier field (CDW1.NSID) set to FFFFFFFFh.
7. Support for the Namespace Identifier field set to FFFFFFFFh is dependant on the Directive Operation (refer to section 9).

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

Figure 13: Opcodes for Admin Commands – NVM Command Set Specific

Opcode (07)	Opcode (06:02)	Opcode (01:00)	Opcode[2]	O/M[1]	Namespace Identifier Used[3]	Command
Generic Command	Function	Data Transfer[4]				
1b	000 00b	00b	80h	O	Yes	Format NVM
1b	000 00b	01b	81h	O	NOTE 5	Security Send
1b	000 00b	10b	82h	O	NOTE 5	Security Receive
1b	000 01b	00b	84h	O	No	Sanitize

NOTES:

1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. A subset of commands uses the Namespace Identifier field (CDW1.NSID). If the Namespace Identifier field is used, then unless otherwise specified, the value FFFFFFFFh is supported in this field. When this field is not used, the field ~~shall be~~ is cleared to 0h as described in Figure 11.
4. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.
5. The use of the Namespace Identifier is Security Protocol specific.

Modify a portion of section 5.3 (Create I/O Completion Queue command) as shown below:

5.3 Create I/O Completion Queue command

…

5.3.1 Command Completion

…

Figure 54: Create I/O Completion Queue – Command Specific Status Values

Value	Description
1h	**Invalid Queue Identifier:** The creation of the I/O Completion Queue failed due to an invalid queue identifier specified as part of the command. An invalid queue identifier is one that ~~is currently in use or one that~~ identifies the Admin Queue (i.e., 0h), is outside the range supported by the controller, or is a Completion Queue Identifier that is already in use.
…	

Modify a portion of section 5.4 (Create I/O Submission Queue command) as shown below:

5.4 Create I/O Submission Queue command

…

Figure 57: Create I/O Submission Queue – Command Dword 11

Bit	Description
31:16	**Completion Queue Identifier (CQID):** This field indicates the identifier of the I/O Completion Queue to utilize for any command completions entries associated with this Submission Queue. ~~The value of 0h (Admin Completion Queue) shall not be specified.~~ If the value specified: a) is 0h (i.e., the Admin Completion Queue) ~~or does not correspond to a valid I/O Completion Queue~~, then the controller should return an error of Invalid Queue Identifier; b) is outside the range supported by the controller, then the controller should return an error of Invalid Queue Identifier; or c) is within the range supported by the controller and does not identify an I/O Completion Queue that has been created, then the controller should return an error of Completion Queue Invalid
…	

5.4.1 Command Completion

~~When the command is completed, the controller posts a completion queue entry to the Admin Completion Queue indicating the status for the command.~~ Upon completion of the Create I/O Submission Queue command, the controller posts a completion queue entry to the Admin Completion Queue.

Create I/O Submission Queue command specific status values are defined in Figure 58.

Figure 58: Create I/O Submission Queue – Command Specific Status Values

Value	Description
0h	**Completion Queue Invalid:** The Completion Queue identifier specified in the command ~~does not exist~~ has not been created.
1h	**Invalid Queue Identifier:** The creation of the I/O Submission Queue failed due an invalid queue identifier specified as part of the command. An invalid queue identifier is one that ~~is currently in use or one that~~ identifies the Admin Queue (i.e., 0h), is outside the range supported by the controller, or is a Submission Queue Identifier that is already in use.
2h	**Invalid Queue Size:** The host attempted to create an I/O Completion Queue with an invalid number of entries (e.g., a value of zero or a value which exceeds the maximum supported by the controller, specified in CAP.MQES).

Modify a portion of section 5.12 (Firmware Image Download command) as shown below:

5.12 Firmware Image Download command

…

The image may be constructed of multiple pieces that are individually downloaded with separate Firmware Image Download commands. Each Firmware Image Download command includes a Dword Offset and Number of Dwords that specify a Dword range. The host software should ensure that image pieces do not have Dword ranges that overlap and that the NUMD field and OFST field meet the alignment and granularity requirements indicated in the FWUG field (refer to Figure 109). Firmware portions may be submitted out of order to the controller. Host software shall submit image portions in order when updating a Boot Partition. If ranges overlap, the controller may return an error of Overlapping Range.

…

Figure 79: Firmware Image Download – Command Dword 10

Bit	Description
31:00	**Number of Dwords (NUMD):** This field specifies the number of Dwords to transfer for this portion of the firmware. This is a 0's based value. If the value specified in this field does not meet the requirement indicated by the FWUG field (refer to Figure 109), the firmwware update may fail with status of Invalid Field in Command.

Figure 80: Firmware Image Download – Command Dword 11

Bit	Description
31:00	**Offset (OFST):** This field specifies the number of Dwords offset from the start of the firmware image being downloaded to the controller. The offset is used to construct the complete firmware image when the firmware is downloaded in multiple pieces. The piece corresponding to the start of the firmware image shall have an Offset of 0h. If the value specified in this field does not meet the requirement indicated by the FWUG field (refer to Figure 109), the firmware update may fail with status of Invalid Field in Command.

Modify a portion of section 5.14.1 (Log Specific Information), 5.14.1.2 (SMART / Health Information), and 5.14.1.3 (Changed Namespace List) as shown below (changes shown here are based on changes published in ECN-004:

5.14 Get Log Page command

…

5.14.1 Log Specific Information

Figure 90 and Figure 91 define the Log pages that may be retrieved with the Get Log Page command **...**

Figure 90: Get Log Page – Log Page Identifiers

Log Identifier	O/M	Scope	Description	Reference Section
00h	Reserved			
01h	M	Controller	Error Information	5.14.1.1
02h	M	Controller [1]	SMART / Health Information	5.14.1.2
	O	Namespace [2]		
03h	M	NVM subsystem	Firmware Slot Information	5.14.1.3
04h	O	Controller	Changed Namespace List	5.14.1.4
05h	O	Controller	Commands Supported and Effects	5.14.1.5
06h	O	Controller [3]	Device Self-test	5.14.1.6
	O	NVM subsystem [4]		
07h	O	Controller	Telemetry Host-Initiated	5.14.1.7
08h	O	Controller	Telemetry Controller-Initiated	5.14.1.8
09h – 6Fh	Reserved			
70h	Discovery (refer to the NVMe over Fabrics specification)			
71h – 7Fh	Reserved for NVMe over Fabrics			
80h – BFh	I/O Command Set Specific			
C0h – FFh	Vendor specific			

KEY:

O = Optional, M = Mandatory
Namespace = The log page contains information about a specific namespace.
Controller = The log page contains information about the controller that is processing the command.
NVM subsystem = The log page contains information about the NVM subsystem.

NOTES:

1. For namespace identifiers of 0h or FFFFFFFFh.
2. For namespace identifiers other than 0h or FFFFFFFFh.
3. Bit 0 is cleared to '0' in the DSTO field in the Identify Controller data structure (refer to Figure 109).
4. Bit 0 is set to '1' in the DSTO field in the Identify Controller data structure.

…

5.14.1.2 SMART / Health Information (Log Identifier 02h)

…

Figure 93: Get Log Page – SMART / Health Information Log

…	
4	**Available Spare Threshold:** When the Available Spare falls below the threshold indicated in this field, an asynchronous event completion may occur. The value is indicated as a normalized percentage (0 to 100%). The values 101 to 255 are reserved.
…	

…

5.14.1.4 Changed Namespace List (Log Identifier 04h)

This log page is used to describe namespaces ~~in~~ attached to this controller that have:

a) changed Identify Namespace information since the last time the log page was read;
b) been added; and
c) been deleted.

The log page is a Namespace List with up to 1024 entries in it. If more than 1024 namespaces have changed attributes since the last time the log page was read, the first entry in the log page shall be set to FFFFFFFFh and the remainder of the list shall be zero-filled.

Replace all of section 5.15 (Identify command) as shown below (changes shown here are based on changes published in ECN-004):

5.15 Identify command

5.15.1 Identify command overview

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

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

Figure 107: Identify – Data Pointer

Bit	Description
127:00	**Data Pointer (DPTR):** This field specifies the start of the data buffer. Refer to Figure for the definition of this field. If using PRPs, this field shall not be a pointer to a PRP List as the data buffer may not cross more than one page boundary.

Figure 108: Identify – Command Dword 10

Bit	Description
31:16	**Controller Identifier (CNTID):** This field specifies the controller identifier used as part of some Identify operations. If the field is not used as part of the Identify operation, then host software shall clear this field to 0h for backwards compatibility (0h is a valid controller identifier). Controllers that support the Namespace Management capability (refer to section 8.12) shall support this field. ~~This field is used for Identify operations with a CNS value of 12h or 13h. This field should be cleared to 0h for Identify operations with a CNS value of 00h, 01h, 02h,10h, and 11h.~~
15:08	Reserved
07:00	**Controller or Namespace Structure (CNS):** This field specifies the information to be returned to the host. Refer to Figure 106.

The data structure returned is based on the Controller or Namespace Structure (CNS) field as shown in Figure 106. If there are fewer entries to return for the data structure indicated based on CNS value, then the unused portion of the list is zero filled. If a controller does not support a CNS value then it shall abort the command with a status of Invalid Field in Command.

Note: The CNS field was specified as a one bit field in revision 1.0 and as a two bit field in revision 1.1. Host software should only issue CNS values defined in revision 1.0 to controllers compliant with revision 1.0. Host software should only issue CNS values defined in revision 1.1 to controllers compliant with revision 1.1. The results of issuing other CNS values to controllers compliant with revision 1.0 or 1.1, respectively, are indeterminate.

The Identify Controller data structure and Identify Namespace data structure include several identifiers. The format and layout of these identifiers is described in section Figure 116.

Figure 106: Identify – CNS Values

CNS value	O/M [1]	Description	NSID[2]	CNTID[3]	Reference Section
Active Namespace Management					
00h	M	Identify Namespace data structure for the specified NSID or the common namespace capabilities	Y	N	5.15.2
01h	M	Identify Controller data structure for the controller processing the command	N	N	5.15.3
02h	M	Active Namespace ID list	Y	N	5.15.4
03h	M	Namespace Identification Descriptor list for the specified NSID	Y	N	5.15.5
04h-0Fh		Reserved			
Controller and Namespace Management					
10h	~~Note 1~~ O [4]	Allocated Namespace ID list	Y	N	5.15.6
11h	~~Note 1~~ O [4]	Identify Namespace data structure for the specified allocated NSID	Y	N	5.15.7
12h	~~Note 1~~ O [4]	Controller identifier list of controllers attached to the specified NSID	Y	Y	5.15.8
13h	~~Note 1~~ O [4]	Controller identifier list of controllers that ~~may or may not be attached to namespace(s)~~ exist in the NVM subsystem.	N	Y	5.15.9
14h	~~Note 2~~ O [5]	Primary Controller Capabilities data structure for the specified primary controller	N	Y	5.15.10
15h	~~Note 2~~ O [5]	Secondary Controller list of controllers associated with the primary controller ~~issuing~~ processing the command	N	Y	5.15.11

CNS value	O/M [1]	Description	NSID[2]	CNTID[3]	Reference Section
16h-1Fh		Reserved			
Future Definition					
20h - FFh		Reserved			

NOTES:

1 O/M definition: O = Optional, M = Mandatory.
2 The CDW1.NSID field is used: Y = Yes, N = No.
3 The CDW10.CNTID field is used: Y = Yes, N = No.
4 Mandatory for controllers that support the Namespace Management capability (refer to section 8.12).
5 Mandatory for controllers that support Virtualization Enhancements (refer to section 8.5).

5.15.2 Identify Namespace data structure (CNS 00h)

The Identify Namespace data structure (refer to Figure 114) is returned to the host for the namespace specified in the Namespace Identifier (CDW1.NSID) field if it is an active NSID. If the specified namespace is not an active NSID, then the controller returns a zero filled data structure.

If the controller supports the Namespace Management capability (refer to section 8.12) and CDW1.NSID is set to FFFFFFFFh then, the controller returns an Identify Namespace data structure that specifies capabilities that are common across namespaces for this controller. If the controller does not support the Namespace Management capability and CDW1.NSID is set to FFFFFFFFh, then the controller shall fail the command with a status code of Invalid Namespace or Format.

Figure 114: Identify – Identify Namespace Data Structure, NVM Command Set Specific

Bytes	O/M [1]	Description
7:0	M	**Namespace Size (NSZE):** This field indicates the total size of the namespace in logical blocks. A namespace of size *n* consists of LBA 0 through (*n* - 1). The number of logical blocks is based on the formatted LBA size. This field is undefined prior to the namespace being formatted.
15:8	M	**Namespace Capacity (NCAP):** This field indicates the maximum number of logical blocks that may be allocated in the namespace at any point in time. The number of logical blocks is based on the formatted LBA size. This field is undefined prior to the namespace being formatted. This field is used in the case of thin provisioning and reports a value that is smaller than or equal to the Namespace Size. Spare LBAs are not reported as part of this field. A logical block is allocated when it is written with a Write or Write Uncorrectable command. A logical block may be deallocated using the Dataset Management, Sanitize, or Write Zeroes command.
23:16	M	**Namespace Utilization (NUSE):** This field indicates the current number of logical blocks allocated in the namespace. This field is smaller than or equal to the Namespace Capacity. The number of logical blocks is based on the formatted LBA size. When using the NVM command set: A logical block is allocated when it is written with a Write or Write Uncorrectable command. A logical block may be deallocated using the Dataset Management, Sanitize, or Write Zeroes command. A controller may report NUSE equal to NCAP at all times if the product is not targeted for thin provisioning environments.

Bytes	O/M [1]	Description
24	M	**Namespace Features (NSFEAT):** This field defines features of the namespace. Bits 7:4 are reserved. Bit 3 if set to '1' indicates that the non-zero NGUID and non-zero EUI64 fields for this namespace are never reused by the controller. If cleared to '0', then the NGUID and EUI64 values 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 7.11. Bit 2 if set to '1' indicates that the controller supports the Deallocated or Unwritten Logical Block error for this namespace. If cleared to '0', then the controller does not support the Deallocated or Unwritten Logical Block error for this namespace. Refer to section 6.7.1.1. Bit 1 if set to '1' indicates that the fields NAWUN, NAWUPF, and NACWU are defined for this namespace and should be used by the host for this namespace instead of the AWUN, AWUPF, and ACWU fields in the Identify Controller data structure. If cleared to '0', then the controller does not support the fields NAWUN, NAWUPF, and NACWU for this namespace. In this case, the host should use the AWUN, AWUPF, and ACWU fields defined in the Identify Controller data structure in Figure 109. Refer to section 6.4. Bit 0 if set to '1' indicates that the namespace supports thin provisioning. Specifically, the Namespace Capacity reported may be less than the Namespace Size. When this feature is supported and the Dataset Management command is supported then deallocating LBAs shall be reflected in the Namespace Utilization field. Bit 0 if cleared to '0' indicates that thin provisioning is not supported and the Namespace Size and Namespace Capacity fields report the same value.
25	M	**Number of LBA Formats (NLBAF):** This field defines the number of supported LBA data size and metadata size combinations supported by the namespace. LBA formats shall be allocated in order (starting with 0) and packed sequentially. This is a 0's based value. The maximum number of LBA formats that may be indicated as supported is 16. The supported LBA formats are indicated in bytes 128 – 191 in this data structure. The LBA Format fields with an index beyond the value set in this field are invalid and not supported. LBA Formats that are valid, but not currently available may be indicated by setting the LBA Data Size for that LBA Format to 0h. The metadata may be either transferred as part of the LBA (creating an extended LBA which is a larger LBA size that is exposed to the application) or it may be transferred as a separate contiguous buffer of data. The metadata shall not be split between the LBA and a separate metadata buffer. It is recommended that software and controllers transition to an LBA size that is 4KB or larger for ECC efficiency at the controller. If providing metadata, it is recommended that at least 8 bytes are provided per logical block to enable use with end-to-end data protection, refer to section 8.2.
26	M	**Formatted LBA Size (FLBAS):** This field indicates the LBA data size & metadata size combination that the namespace has been formatted with (refer to section 5.23). Bits 7:5 are reserved. Bit 4 if set to '1' indicates that the metadata is transferred at the end of the data LBA, creating an extended data LBA. Bit 4 if cleared to '0' indicates that all of the metadata for a command is transferred as a separate contiguous buffer of data. Bit 4 is not applicable when there is no metadata. Bits 3:0 indicates one of the 16 supported LBA Formats indicated in this data structure.

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td>27</td><td>M</td><td>Metadata Capabilities (MC): This field indicates the capabilities for metadata.

Bits 7:2 are reserved.

Bit 1 if set to ‘1’ indicates the namespace supports the metadata being transferred as part of a separate buffer that is specified in the Metadata Pointer. Bit 1 if cleared to ‘0’ indicates that the namespace does not support the metadata being transferred as part of a separate buffer.

Bit 0 if set to ‘1’ indicates that the namespace supports the metadata being transferred as part of an extended data LBA. Bit 0 if cleared to ‘0’ indicates that the namespace does not support the metadata being transferred as part of an extended data LBA.</td></tr>
<tr><td>28</td><td>M</td><td>End-to-end Data Protection Capabilities (DPC): This field indicates the capabilities for the end-to-end data protection feature. Multiple bits may be set in this field. Refer to section 8.3.

Bits 7:5 are reserved.

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.

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.

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.

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.

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.</td></tr>
<tr><td>29</td><td>M</td><td>End-to-end Data Protection Type Settings (DPS): This field indicates the Type settings for the end-to-end data protection feature. Refer to section 8.3.

Bits 7:4 are reserved.

Bit 3 if set to ‘1’ indicates that the protection information, if enabled, is transferred as the first eight bytes of metadata. Bit 3 if cleared to ‘0’ indicates that the protection information, if enabled, is transferred as the last eight bytes of metadata.

Bits 2:0 indicate whether Protection Information is enabled and the type of Protection Information enabled. The values for this field have the following meanings:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Protection information is not enabled</td></tr>
<tr><td>001b</td><td>Protection information is enabled, Type 1</td></tr>
<tr><td>010b</td><td>Protection information is enabled, Type 2</td></tr>
<tr><td>011b</td><td>Protection information is enabled, Type 3</td></tr>
<tr><td>100b – 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
</table>

Bytes	O/M [1]	Description
30	O	**Namespace Multi-path I/O and Namespace Sharing Capabilities (NMIC):** This field specifies multi-path I/O and namespace sharing capabilities of the namespace. Bits 7:1 are reserved. Bit 0: 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 may only be attached to one controller at a time.
31	O	**Reservation Capabilities (RESCAP):** This field indicates the reservation capabilities of the namespace. A value of 00h in this field indicates that reservations are not supported by this namespace. Refer to section 8.8 for more details. 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. 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. 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. 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. 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. 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. 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. 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.
32	O	**Format Progress Indicator (FPI):** If a format operation is in progress, this field indicates the percentage of the namespace that remains to be formatted. Bit 7 if set to '1' indicates that the namespace supports the Format Progress Indicator defined by bits 6:0 in this field. If this bit is cleared to '0', then the namespace does not support the Format Progress Indicator and bits 6:0 in this field shall be cleared to 0h. Bits 6:0 indicate the percentage of the Format NVM command that remains to be completed (e.g., a value of 25 indicates that 75% of the Format NVM command has been completed and 25% remains to be completed). If bit 7 is set to '1', then a value of 0 indicates that the namespace is formatted with the format specified by the FLBAS and DPS fields in this data structure and there is no Format NVM command in progress.

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td>33</td><td>O</td><td>Deallocate Logical Block Features (DLFEAT): This field indicates information about features that affect deallocating logical blocks for this namespace.

Bits 7:5 are reserved.

Bit 4 if set to ‘1’ indicates that the Guard field for deallocated logical blocks that contain protection information is set to the CRC for the value read from the deallocated logical block and its metadata (excluding protection information). If cleared to ‘0’ indicates that the Guard field for the deallocated logical blocks that contain protection information is set to FFFFh.

Bit 3 if set to ‘1’ indicates that the controller supports the Deallocate bit in the Write Zeroes command for this namespace. If cleared to ‘0’ indicates that the controller does not support the Deallocate bit in the Write Zeroes command for this namespace. This bit shall be set to the same value for all namespaces in the NVM subsystem.

Bits 2:0 indicate the values read from a deallocated logical block and its metadata (excluding protection information). The values for this field have the following meanings:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Not reported</td></tr>
<tr><td>001b</td><td>All bytes set to 00h</td></tr>
<tr><td>010b</td><td>All bytes set to FFh</td></tr>
<tr><td>011b – 111b</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>35:34</td><td>O</td><td>Namespace Atomic Write Unit Normal (NAWUN): This field indicates the namespace specific size of the write operation guaranteed to be written atomically to the NVM during normal operation.

A value of 0h indicates that the size for this namespace is the same size as that reported in the AWUN field of the Identify Controller data structure. All other values specify a size in terms of logical blocks using the same encoding as the AWUN field. Refer to section 6.4.</td></tr>
<tr><td>37:36</td><td>O</td><td>Namespace Atomic Write Unit Power Fail (NAWUPF): This field indicates the namespace specific size of the write operation guaranteed to be written atomically to the NVM during a power fail or error condition.

A value of 0h indicates that the size for this namespace is the same size as that reported in the AWUPF field of the Identify Controller data structure. All other values specify a size in terms of logical blocks using the same encoding as the AWUPF field. Refer to section 6.4.</td></tr>
<tr><td>39:38</td><td>O</td><td>Namespace Atomic Compare & Write Unit (NACWU): This field indicates the namespace specific size of the write operation guaranteed to be written atomically to the NVM for a Compare and Write fused command.

A value of 0h indicates that the size for this namespace is the same size as that reported in the ACWU field of the Identify Controller data structure. All other values specify a size in terms of logical blocks using the same encoding as the ACWU field. Refer to section 6.4.</td></tr>
<tr><td>41:40</td><td>O</td><td>Namespace Atomic Boundary Size Normal (NABSN): This field indicates the atomic boundary size for this namespace for the NAWUN value. This field is specified in logical blocks. Writes to this namespace that cross atomic boundaries are not guaranteed to be atomic to the NVM with respect to other read or write commands.

A value of 0h indicates that there are no atomic boundaries for normal write operations. All other values specify a size in terms of logical blocks using the same encoding as the AWUN field. Refer to section 6.4.</td></tr>
</table>

Bytes	O/M [1]	Description
43:42	O	**Namespace Atomic Boundary Offset (NABO):** This field indicates the LBA on this namespace where the first atomic boundary starts. If the NABSN and NABSPF fields are cleared to 0h, then the NABO field shall be cleared to 0h. NABO shall be less than or equal to NABSN and NABSPF. Refer to section 6.4.
45:44	O	**Namespace Atomic Boundary Size Power Fail (NABSPF):** This field indicates the atomic boundary size for this namespace specific to the Namespace Atomic Write Unit Power Fail value. This field is specified in logical blocks. Writes to this namespace that cross atomic boundaries are not guaranteed to be atomic with respect to other read or write commands and there is no guarantee of data returned on subsequent reads of the associated logical blocks. A value of 0h indicates that there are no atomic boundaries for power fail or error conditions. All other values specify a size in terms of logical blocks using the same encoding as the AWUPF field. Refer to section 6.4.
47:46	O	**Namespace Optimal IO Boundary (NOIOB):** This field indicates the optimal IO boundary for this namespace. This field is specified in logical blocks. The host should construct read and write commands that do not cross the IO boundary to achieve optimal performance. A value of 0h indicates that no optimal IO boundary is reported.
63:48	O	**NVM Capacity (NVMCAP):** This field indicates the total size of the NVM allocated to this namespace. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to section 8.12) is ~~and Namespace Attachment commands are~~ supported. Note: This field may not correspond to the logical block size multiplied by the Namespace Size field. Due to thin provisioning or other settings (e.g., endurance), this field may be larger or smaller than the Namespace Size reported.
103:64		Reserved
119:104	O	**Namespace Globally Unique Identifier (NGUID):** This field contains a 128-bit value that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., controller reset, namespace format, etc.). This field uses the EUI-64 based 16-byte designator format. Bytes 114:112 contain the 24-bit Organizationally Unique Identifier (OUI) value assigned by the IEEE Registration Authority. Bytes 119:115 contain an extension identifer assigned by the corresponding organization. Bytes 111:104 contain the vendor specific extension identifier assigned by the corresponding organization. See the IEEE EUI-64 guidelines for more information. This field is big endian (refer to section 7.10). The controller shall specify a globally unique namespace identifier in this field, the EUI64 field, or a Namespace UUID in the Namespace Identification Descriptor (refer to Figure 116) when the namespace is created. If the controller is not able to provide a globally unique identifier in this field, then this field shall be cleared to 0h. Refer to section 7.11.
127:120	O	**IEEE Extended Unique Identifier (EUI64):** This field contains a 64-bit IEEE Extended Unique Identifier (EUI-64) that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., controller reset, namespace format, etc.). The EUI-64 is a concatenation of a 24-bit or 36-bit Organizationally Unique Identifier (OUI or OUI-36) value assigned by the IEEE Registration Authority and an extension identifier assigned by the corresponding organization. See the IEEE EUI-64 guidelines for more information. This field is big endian (refer to section 7.10). The controller shall specify a globally unique namespace identifier in this field, the NGUID field, or a Namespace UUID in the Namespace Identification Descriptor (refer to Figure 116) when the namespace is created. If the controller is not able to provide a globally unique 64-bit identifier in this field, then this field shall be cleared to 0h. Refer to section 7.11.

Bytes	O/M [1]	Description
131:128	M	**LBA Format 0 Support (LBAF0):** This field indicates the LBA format 0 that is supported by the controller. The LBA format field is defined in Figure 115.
135:132	O	**LBA Format 1 Support (LBAF1):** This field indicates the LBA format 1 that is supported by the controller. The LBA format field is defined in Figure 115.
139:136	O	**LBA Format 2 Support (LBAF2):** This field indicates the LBA format 2 that is supported by the controller. The LBA format field is defined in Figure 115.
143:140	O	**LBA Format 3 Support (LBAF3):** This field indicates the LBA format 3 that is supported by the controller. The LBA format field is defined in Figure 115.
147:144	O	**LBA Format 4 Support (LBAF4):** This field indicates the LBA format 4 that is supported by the controller. The LBA format field is defined in Figure 115.
151:148	O	**LBA Format 5 Support (LBAF5):** This field indicates the LBA format 5 that is supported by the controller. The LBA format field is defined in Figure 115.
155:152	O	**LBA Format 6 Support (LBAF6):** This field indicates the LBA format 6 that is supported by the controller. The LBA format field is defined in Figure 115.
159:156	O	**LBA Format 7 Support (LBAF7):** This field indicates the LBA format 7 that is supported by the controller. The LBA format field is defined in Figure 115.
163:160	O	**LBA Format 8 Support (LBAF8):** This field indicates the LBA format 8 that is supported by the controller. The LBA format field is defined in Figure 115.
167:164	O	**LBA Format 9 Support (LBAF9):** This field indicates the LBA format 9 that is supported by the controller. The LBA format field is defined in Figure 115.
171:168	O	**LBA Format 10 Support (LBAF10):** This field indicates the LBA format 10 that is supported by the controller. The LBA format field is defined in Figure 115.
175:172	O	**LBA Format 11 Support (LBAF11):** This field indicates the LBA format 11 that is supported by the controller. The LBA format field is defined in Figure 115.
179:176	O	**LBA Format 12 Support (LBAF12):** This field indicates the LBA format 12 that is supported by the controller. The LBA format field is defined in Figure 115.
183:180	O	**LBA Format 13 Support (LBAF13):** This field indicates the LBA format 13 that is supported by the controller. The LBA format field is defined in Figure 115.
187:184	O	**LBA Format 14 Support (LBAF14):** This field indicates the LBA format 14 that is supported by the controller. The LBA format field is defined in Figure 115.
191:188	O	**LBA Format 15 Support (LBAF15):** This field indicates the LBA format 15 that is supported by the controller. The LBA format field is defined in Figure 115.
383:192		Reserved
4095:384	O	Vendor Specific

NOTES:

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

The LBA format data structure is described in Figure 115.

Figure 115: Identify – LBA Format Data Structure, NVM Command Set Specific

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:26</td><td>Reserved</td></tr>
<tr><td>25:24</td><td>Relative Performance (RP): This field indicates the relative performance of the LBA format indicated relative to other LBA formats supported by the controller. Depending on the size of the LBA and associated metadata, there may be performance implications. The performance analysis is based on better performance on a queue depth 32 with 4KB read workload. The meanings of the values indicated are included in the following table.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>Best performance</td></tr>
<tr><td>01b</td><td>Better performance</td></tr>
<tr><td>10b</td><td>Good performance</td></tr>
<tr><td>11b</td><td>Degraded performance</td></tr>
</table></td></tr>
<tr><td>23:16</td><td>LBA Data Size (LBADS): This field indicates the LBA data size supported. The value is reported in terms of a power of two (2^n). A value smaller than 9 (i.e. 512 bytes) is not supported. If the value reported is 0h then the LBA format is not supported / used or is not currently available.</td></tr>
<tr><td>15:00</td><td>Metadata Size (MS): This field indicates the number of metadata bytes provided per LBA based on the LBA Data Size indicated. If there is no metadata supported, then this field shall be cleared to 00h.

If metadata is supported, then the namespace may support the metadata being transferred as part of an extended data LBA or as part of a separate contiguous buffer. If end-to-end data protection is enabled, then the first eight bytes or last eight bytes of the metadata is the protection information.</td></tr>
</table>

5.15.3 Identify Controller data structure (CNS 01h)

The Identify Controller data structure (refer to Figure 109) is returned to the host for this controller.

Figure 109: Identify – Identify Controller Data Structure

Bytes	O/M [1]	Description
		Controller Capabilities and Features
01:00	M	**PCI Vendor ID (VID):** Contains the company vendor identifier that is assigned by the PCI SIG. This is the same value as reported in the ID register in section 2.1.1.
03:02	M	**PCI Subsystem Vendor ID (SSVID):** Contains the company vendor identifier that is assigned by the PCI SIG for the subsystem. This is the same value as reported in the SS register in section 2.1.17.
23:04	M	**Serial Number (SN):** Contains the serial number for the NVM subsystem that is assigned by the vendor as an ASCII string. Refer to section 7.10 for unique identifier requirements. Refer to section 1.5 for ASCII string requirements.
63:24	M	**Model Number (MN):** Contains the model number for the NVM subsystem that is assigned by the vendor as an ASCII string. Refer to section 7.10 for unique identifier requirements. Refer to section 1.5 for ASCII string requirements.
71:64	M	**Firmware Revision (FR):** Contains the currently active firmware revision for the NVM subsystem. This is the same revision information that may be retrieved with the Get Log Page command, refer to section 5.14.1.3. Refer to section 1.5 for ASCII string requirements.
72	M	**Recommended Arbitration Burst (RAB):** This is the recommended Arbitration Burst size. The value is in commands and is reported as a power of two (2^n). This is the same units as the Arbitration Burst size. Refer to section 4.11.
75:73	M	**IEEE OUI Identifier (IEEE):** Contains the Organization Unique Identifier (OUI) for the controller vendor. The OUI shall be a valid IEEE/RAC assigned identifier that may be registered at http://standards.ieee.org/develop/regauth/oui/public.html.

Bytes	O/M [1]	Description
76	O	**Controller Multi-Path I/O and Namespace Sharing Capabilities (CMIC):** This field specifies multi-path I/O and namespace sharing capabilities of the controller and NVM subsystem. Bits 7:3 are reserved Bit 2: If set to '1' then the controller is associated with an SR-IOV Virtual Function. If cleared to '0' then the controller is associated with a PCI Function or a Fabrics connection. Bit 1: If set to '1' then the NVM subsystem may contain two or more controllers. If cleared to '0' then the NVM subsystem contains only a single controller. As described in section 1.4.1 an NVM subsystem that contains multiple controllers may be used by multiple hosts, or may provide multiple paths for a single host. Bit 0: If set to '1' then the NVM subsystem may contain more than one NVM subsystem port. If cleared to '0' then the NVM subsystem contains only a single NVM subsystem port.
77	M	**Maximum Data Transfer Size (MDTS):** This field indicates the maximum data transfer size between the host and the controller. The host should not submit a command that exceeds this transfer size. If a command is submitted that exceeds the transfer size, then the command is aborted with a status of Invalid Field in Command. The value is in units of the minimum memory page size (CAP.MPSMIN) and is reported as a power of two (2^*n*). A value of 0h indicates no restrictions on transfer size. The restriction includes metadata if it is interleaved with the logical block data. The restriction does not apply to commands that do not transfer data between the host and the controller (e.g., Write Uncorrectable command or Write Zeroes command). If SGL Bit Bucket descriptors are supported, their lengths shall be included in determining if a command exceeds the Maximum Data Transfer Size for destination data buffers. Their length in a source data buffer is not included for a Maximum Data Transfer Size calculation.
79:78	M	**Controller ID (CNTLID):** Contains the NVM subsystem unique controller identifier associated with the controller. Refer to section 7.10 for unique identifier requirements.
83:80	M	**Version (VER):** This field contains the value reported in the Version register defined in section 3.1.2. Implementations compliant to revision 1.2 or later of this specification shall report a non-zero value in this field.
87:84	M	**RTD3 Resume Latency (RTD3R):** This field indicates the expected latency in microseconds to resume from Runtime D3 (RTD3). Refer to section 8.4.4. A value of 0h indicates RTD3 Resume Latency is not reported.
91:88	M	**RTD3 Entry Latency (RTD3E):** This field indicates the typical latency in microseconds to enter Runtime D3 (RTD3). Refer to section 8.4.4 ~~for test conditions~~. A value of 0h indicates RTD3 Entry Latency is not reported.
95:92	M	**Optional Asynchronous Events Supported (OAES):** This field indicates the optional asynchronous events supported by the controller. A controller shall not send optional asynchronous events before they are enabled by host software. Bits 31:10 are reserved. Bit 9 is set to '1' if the controller supports sending Firmware Activation Notices. If cleared to '0' then the controller does not support the Firmware Activation Notices event. Bit 8 is set to '1' if the controller supports sending Namespace Attribute Notices and the associated Changed Namespace List log page. If cleared to '0' then the controller does not support the Namespace Attribute Notices event nor the associated Changed Namespace List log page. Bits 7:0 are reserved.

Bytes	O/M [1]	Description
99:96	M	**Controller Attributes (CTRATT):** This field indicates attributes of the controller. Bits 31:2 are reserved. Bit 1 (Non-Operational Power State Permissive Mode): If set to '1' then the controller supports host control of whether the controller may temporarily exceed the power of a non-operational power state for the purpose of executing controller initiated background operations in a non-operational power state (i.e., Non-Operational Power State Permissive Mode supported). If cleared to '0' then the controller does not support host control of whether the controller may exceed the power of a non-operational state for the purpose of executing controller initiated background operations in a non-operational state (i.e., Non-Operational Power State Permissive Mode not supported). Refer to section 5.21.1.17. Bit 0 if set to '1' then the controller supports a 128-bit Host Identifier. Bit 0 if cleared to '0' then the controller does not support a 128-bit Host Identifier.
111:100		Reserved
127:112	O	**FRU Globally Unique Identifier (FGUID):** This field contains a 128-bit value that is globally unique for a given Field Replaceable Unit (FRU). Refer to the NVM Express Management Interface (NVMe-MI) specification for the definition of a FRU. This field remains fixed throughout the life of the FRU. This field shall contain the same value for each controller associated with a given FRU. This field uses the EUI-64 based 16-byte designator format. Bytes 122:120 contain the 24-bit Organizationally Unique Identifier (OUI) value assigned by the IEEE Registration Authority. Bytes 127:123 contain an extension identifier assigned by the corresponding organization. Bytes 119:112 contain the vendor specific extension identifier assigned by the corresponding organization. See the IEEE EUI-64 guidelines for more information. This field is big endian (refer to section 7.10). When not implemented, this field contains a value of 0h.
239:128		Reserved
255:240		Refer to the NVMe Management Interface Specification for definition.
Admin Command Set Attributes & Optional Controller Capabilities		

Bytes	O/M [1]	Description
257:256	M	**Optional Admin Command Support (OACS):** This field indicates the optional Admin commands and features supported by the controller. Refer to section 5. Bits 15:9 are reserved. Bit 8 if set to '1' then the controller supports the Doorbell Buffer Config command. If cleared to '0' then the controller does not support the Doorbell Buffer Config command. Bit 7 if set to '1' then the controller supports the Virtualization Management command. If cleared to '0' then the controller does not support the Virtualization Management command. Bit 6 if set to '1' then the controller supports the NVMe-MI Send and NVMe-MI Receive commands. If cleared to '0' then the controller does not support the NVMe-MI Send and NVMe-MI Receive commands. Bit 5 if set to '1' then the controller supports Directives. If cleared to '0' then the controller does not support Directives. A controller that supports Directives shall support the Directive Send and Directive Receive commands. Refer to section 9. Bit 4 if set to '1' then the controller supports the Device Self-test command. If cleared to '0' then the controller does not support the Device Self-test command. ~~Bit 3 if set to '1' then the controller supports the Namespace Management and Namespace Attachment commands. If cleared to '0' then the controller does not support the Namespace Management and Namespace Attachment commands.~~ Bit 3 if set to '1' then the controller supports the Namespace Management capability (refer to section 8.12). If cleared to '0' then the controller does not support the Namespace Management capability. Bit 2 if set to '1' then the controller supports the Firmware Commit and Firmware Image Download commands. If cleared to '0' then the controller does not support the Firmware Commit and Firmware Image Download commands. Bit 1 if set to '1' then the controller supports the Format NVM command. If cleared to '0' then the controller does not support the Format NVM command. Bit 0 if set to '1' then the controller supports the Security Send and Security Receive commands. If cleared to '0' then the controller does not support the Security Send and Security Receive commands.
258	M	**Abort Command Limit (ACL):** This field is used to convey the maximum number of concurrently executing Abort commands supported by the controller (refer to section 5.1. This is a 0's based value. It is recommended that implementations support concurrent execution of a minimum of four Abort commands.
259	M	**Asynchronous Event Request Limit (AERL):** This field is used to convey the maximum number of concurrently outstanding Asynchronous Event Request commands supported by the controller (see section 5.2). This is a 0's based value. It is recommended that implementations support a minimum of four Asynchronous Event Request Limit commands outstanding simultaneously.

Bytes	O/M [1]	Description
260	M	**Firmware Updates (FRMW):** This field indicates capabilities regarding firmware updates. Refer to section 8.1 for more information on the firmware update process. Bits 7:5 are reserved. Bit 4 if set to '1' indicates that the controller supports firmware activation without a reset. If cleared to '0' then the controller requires a reset for firmware to be activated. Bits 3:1 indicate the number of firmware slots that the controller supports. This field shall specify a value between one and seven, indicating that at least one firmware slot is supported and up to seven maximum. This corresponds to firmware slots 1 through 7. Bit 0 if set to '1' indicates that the first firmware slot (slot 1) is read only. If cleared to '0' then the first firmware slot (slot 1) is read/write. Implementations may choose to have a baseline read only firmware image.
261	M	**Log Page Attributes (LPA):** This field indicates optional attributes for log pages that are accessed via the Get Log Page command. Bits 7:4 are reserved. Bit 3 if set to '1' then the controller supports the Telemetry Host-Initiated and Telemetry Controller-Initiated log pages and sending Telemetry Log Notices. If cleared to '0' then the controller does not support the Telemetry Host-Initiated and Telemetry Controller-Initiated log pages and Telemetry Log Notice events. Bit 2 if set to '1' then the controller supports extended data for Get Log Page (including extended Number of Dwords and Log Page Offset fields). Bit 2 if cleared to '0' then the controller does not support extended data for Get Log Page. Bit 1 if set to '1' then the controller supports the Commands Supported and Effects log page. Bit 1 if cleared to '0' then the controller does not support the Commands Supported and Effects log page. Bit 0 if set to '1' then the controller supports the SMART / Health information log page on a per namespace basis. If cleared to '0' then the controller does not support the SMART / Health information log page on a per namespace basis.
262	M	**Error Log Page Entries (ELPE):** This field indicates the maximum number of Error Information log entries that are stored by the controller. This field is a 0's based value.
263	M	**Number of Power States Support (NPSS):** This field indicates the number of NVM Express power states supported by the controller. This is a 0's based value. Refer to section 8.4. Power states are numbered sequentially starting at power state 0. A controller shall support at least one power state (i.e., power state 0) and may support up to 31 additional power states (i.e., up to 32 total).
264	M	**Admin Vendor Specific Command Configuration (AVSCC):** This field indicates the configuration settings for Admin Vendor Specific command handling. Refer to section 8.7. Bits 7:1 are reserved. Bit 0 if set to '1' indicates that all Admin Vendor Specific Commands use the format defined in Figure 12. If cleared to '0' indicates that the format of all Admin Vendor Specific Commands are vendor specific.

Bytes	O/M [1]	Description
265	O	**Autonomous Power State Transition Attributes (APSTA):** This field indicates the attributes of the autonomous power state transition feature. Refer to section 8.4.2. Bits 7:1 are reserved. Bit 0 if set to '1' then the controller supports autonomous power state transitions. If cleared to '0' then the controller does not support autonomous power state transitions.
267:266	M	**Warning Composite Temperature Threshold (WCTEMP):** This field indicates the minimum Composite Temperature field value (reported in the SMART / Health Information log in Figure 93) that indicates an overheating condition during which controller operation continues. Immediate remediation is recommended (e.g., additional cooling or workload reduction). The platfom should strive to maintain a composite temperature below this value. A value of 0h in this field indicates that no warning temperature threshold value is reported by the controller. Implementations compliant to revision 1.2 or later of this specification shall report a non-zero value in this field. It is recommended that implementations report a value of 0157h in this field.
269:268	M	**Critical Composite Temperature Threshold (CCTEMP):** This field indicates the minimum Composite Temperature field value (reported in the SMART / Health Information log in Figure 93) that indicates a critical overheating condition (e.g., may prevent continued normal operation, possibility of data loss, automatic device shutdown, extreme peformance throttling, or permanent damage). A value of 0h in this field indicates that no critical temperature threshold value is reported by the controller. Implementations compliant to revision 1.2 or later of this specification shall report a non-zero value in this field.
271:270	O	**Maximum Time for Firmware Activation (MTFA):** Indicates the maximum time the controller temporarily stops processing commands to activate the firmware image. This field shall be valid if the controller supports firmware activation without a reset. This field is specified in 100 millisecond units. A value of 0h indicates that the maximum time is undefined.
275:272	O	**Host Memory Buffer Preferred Size (HMPRE):** This field indicates the preferred size that the host is requested to allocate for the Host Memory Buffer feature in 4KB units. This value shall be larger than or equal to the Host Memory Buffer Minimum Size. If this field is non-zero, then the Host Memory Buffer feature is supported. If this field is cleared to 0h, then the Host Memory Buffer feature is not supported.
279:276	O	**Host Memory Buffer Minimum Size (HMMIN):** This field indicates the minimum size that the host is requested to allocate for the Host Memory Buffer feature in 4KB units. If this field is cleared to 0h, then the host is requested to allocate any amount of host memory possible up to the HMPRE value.
295:280	O	**Total NVM Capacity (TNVMCAP):** This field indicates the total NVM capacity in the NVM subsystem. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to section 8.12) is ~~and Namespace Attachment commands are~~ supported.
311:296	O	**Unallocated NVM Capacity (UNVMCAP):** This field indicates the unallocated NVM capacity in the NVM subsystem. The value is in bytes. This field shall be supported if the Namespace Management capability (refer to section 8.12) is ~~and Namespace Attachment commands are~~ supported.

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td>315:312</td><td>O</td><td>Replay Protected Memory Block Support (RPMBS): This field indicates if the controller supports one or more Replay Protected Memory Blocks (RPMBs) and the capabilities. Refer to section 8.10.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:24</td><td>Access Size: If the Number of RPMB Units field is non-zero, then this field indicates the number of 512B units of data that may be read or written per RPMB access by Security Send or Security Receive commands for this controller. This is a 0’s based value. A value of 0h indicates support for one unit of 512B of data.

If the Number of RPMB Units field is 0h, this field shall be ignored.</td></tr>
<tr><td>23:16</td><td>Total Size: If the Number of RPMB Units field is non-zero, then this field indicates the number of 128KB units of data in each RPMB supported in the controller. This is a 0’s based value. A value of 0h indicates support for one unit of 128KB of data.

If the Number of RPMB Units field is 0h, this field shall be ignored.</td></tr>
<tr><td>15:06</td><td>Reserved</td></tr>
<tr><td>05:03</td><td>Authentication Method: This field indicates the authentication method used to access all RPMBs in the controller. The values for this field are:

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>HMAC SHA-256 (refer to RFC 6234)</td></tr>
<tr><td>001b-111b</td><td>Reserved</td></tr>
</table>
</td></tr>
<tr><td>02:00</td><td>Number of RPMB Units: This field indicates the number of RPMB targets the controller supports. All RPMB targets supported shall have the same capabilities as defined in the RPMBS field. A value of 0h indicates the controller does not support Replay Protected Memory Blocks. If this value is non-zero, then the controller shall support the Security Send and Security Receive commands.</td></tr>
</table>
</td></tr>
<tr><td>317:316</td><td>O</td><td>Extended Device Self-test Time (EDSTT): If the Device Self-test command is supported, then this field indicates the nominal amount of time in one minute units that the controller takes to complete an extended device self-test operation when in power state 0. If the Device Self-test command is not supported, then this field is reserved.</td></tr>
<tr><td>318</td><td>O</td><td>Device Self-test Options (DSTO): This field indicates the optional Device Self-test command or operation behaviors supported by the controller or NVM subsystem.

Bit 0 if set to ‘1’ then the NVM subsystem supports only one device self-test operation in progress at a time. If cleared to ‘0’ then the NVM subsystem supports one device self-test operation per controller at a time.</td></tr>
</table>

Bytes	O/M [1]	Description
319	M	**Firmware Update Granularity (FWUG):** ~~This field indicates the minimum granularity and alignment of the data provided in the Firmware Image Download command. If the data in the Firmware Image Download command does not conform to these granularity and alignment requirements, the firmware update may fail. For the broadest interoperability with software, it is recommended that the controller set this value to the lowest value possible.~~ This field indicates the granularity and alignment requirement of the firmware image being updated by the Firmware Image Download command (refer to section 5.12). If the values specified in the NUMD field or the OFST field in the Firmware Image Download command do not conform to this granularity and alignment requirement, then the firmware update may fail with status of Invalid Field in Command. For the broadest interoperability with host software, it is recommended that the controller set this value to the lowest value possible. The value is reported in 4KB units (e.g., 1h corresponds to 4KB, 2h corresponds to 8KB~~, etc.~~). A value of 0h indicates that no information on granularity is provided. A value of FFh indicates there is no restriction (i.e., any granularity and alignment in Dwords is allowed).
321:320	M	**Keep Alive Support (KAS):** This field indicates the granularity of the Keep Alive Timer in 100 ms units (refer to section 7.12). If this field is cleared to 0h then Keep Alive is not supported. Keep Alive shall be supported for NVMe over Fabrics implementations.
323:322	O	**Host Controlled Thermal Management Attributes (HCTMA):** This field indicates the attributes of the host controlled thermal management feature. Refer to section 8.4.5. Bits 15:1 are reserved. Bit 0 if set to '1' then the controller supports host controlled thermal management. If cleared to '0' then the controller does not support host controlled thermal management. If this bit is set to '1', then ~~then,~~ the controller shall support the Set Features command and Get Features command with the Feature Identifier field set to 10h.
325:324	O	**Minimum Thermal Management Temperature (MNTMT):** This field indicates the minimum temperature, in degrees Kelvin, that the host may request in the Thermal Management Temperature 1 field and Thermal Management Temperature 2 field of a Set Features command with the Feature Identifier field set to 10h. A value of 0000h indicates that the controller does not report this field or the host controlled thermal management feature (refer to section 8.4.5) is not supported.
327:326	O	**Maximum Thermal Management Temperature (MXTMT):** This field indicates the maximum temperature, in degrees Kelvin, that the host may request in the Thermal Management Temperature 1 field and Thermal Management Temperature 2 field of the Set Features command with the Feature Identifier set to 10h. A value of 0000h indicates that the controller does not report this field or the host controlled thermal management feature is not supported.
331:328	O	**Sanitize Capabilities (SANICAP):** This field indicates attributes for sanitize operations. If the Sanitize command is supported then this field shall be non-zero. If the Sanitize command is not supported, then this field ~~is reserved~~ shall be cleared to '0'. Refer to section 8.15. Bits 31:3 are reserved. Bit 2 if set to '1' then the controller supports the Overwrite sanitize operation. If cleared to '0' then the controller does not support the Overwrite sanitize operation. Bit 1 if set to '1' then the controller supports the Block Erase sanitize operation. If cleared to '0' then the controller does not support the Block Erase sanitize operation. Bit 0 if set to '1' then the controller supports the Crypto Erase sanitize operation. If cleared to '0' then the controller does not support the Crypto Erase sanitize operation.

Bytes	O/M [1]	Description
511:332		Reserved
NVM Command Set Attributes		
512	M	**Submission Queue Entry Size (SQES):** This field defines the required and maximum Submission Queue entry size when using the NVM Command Set. Bits 7:4 define the maximum Submission Queue entry size when using the NVM Command Set. This value is larger than or equal to the required SQ entry size. The value is in bytes and is reported as a power of two (2^*n*). The recommended value is 6, corresponding to a standard NVM Command Set SQ entry size of 64 bytes. Controllers that implement proprietary extensions may support a larger value. Bits 3:0 define the required Submission Queue Entry size when using the NVM Command Set. This is the minimum entry size that may be used. The value is in bytes and is reported as a power of two (2^*n*). The required value shall be 6, corresponding to 64.
513	M	**Completion Queue Entry Size (CQES):** This field defines the required and maximum Completion Queue entry size when using the NVM Command Set. Bits 7:4 define the maximum Completion Queue entry size when using the NVM Command Set. This value is larger than or equal to the required CQ entry size. The value is in bytes and is reported as a power of two (2^*n*). The recommended value is 4, corresponding to a standard NVM Command Set CQ entry size of 16 bytes. Controllers that implement proprietary extensions may support a larger value. Bits 3:0 define the required Completion Queue entry size when using the NVM Command Set. This is the minimum entry size that may be used. The value is in bytes and is reported as a power of two (2^*n*). The required value shall be 4, corresponding to 16.
515:514	M	**Maximum Outstanding Commands (MAXCMD):** Indicates the maximum number of commands that the controller processes at one time for a particular queue (which may be larger than the size of the corresponding Submission Queue). The host may use this value to size Completion Queues and optimize the number of commands submitted at one time to a particular I/O Queue. This field is mandatory for NVMe over Fabrics and optional for NVMe over PCIe implementations. If the field is not used, it shall be cleared to 0h.
519:516	M	**Number of Namespaces (NN):** This field defines the maximum number of namespaces supported by the controller. This field also represents the maximum value of a valid NSID for the controller.

Bytes	O/M [1]	Description
521:520	M	**Optional NVM Command Support (ONCS):** This field indicates the optional NVM commands and features supported by the controller. Refer to section 6. Bits 15:7 are reserved. Bit 6 if set to '1' then the controller supports the Timestamp feature. If cleared to '0', then the controller does not support the Timestamp feature. Refer to section 5.21.1.14. Bit 5 if set to '1' then the controller supports reservations. If cleared to '0' then the controller does not support reservations. If the controller supports reservations, then it shall support the following commands associated with reservations: Reservation Report, Reservation Register, Reservation Acquire, and Reservation Release. Refer to section 8.8 for additional requirements. Bit 4 if set to '1' then the controller supports the Save field set to a non-zero value in the Set Features command and the Select field set to a non-zero value in the Get Features command. If cleared to '0' then the controller does not support the Save field set to a non-zero value in the Set Features command and the Select field set to a non-zero value in the Get Features command. Bit 3 if set to '1' then the controller supports the Write Zeroes command. If cleared to '0' then the controller does not support the Write Zeroes command. Bit 2 if set to '1' then the controller supports the Dataset Management command. If cleared to '0' then the controller does not support the Dataset Management command. Bit 1 if set to '1' then the controller supports the Write Uncorrectable command. If cleared to '0' then the controller does not support the Write Uncorrectable command. Bit 0 if set to '1' then the controller supports the Compare command. If cleared to '0' then the controller does not support the Compare command.
523:522	M	**Fused Operation Support (FUSES):** This field indicates the fused operations that the controller supports. Refer to section 6.2. Bits 15:1 are reserved. Bit 0 if set to '1' then the controller supports the Compare and Write fused operation. If cleared to '0' then the controller does not support the Compare and Write fused operation. Compare shall be the first command in the sequence.

Bytes	O/M [1]	Description
524	M	**Format NVM Attributes (FNA):** This field indicates attributes for the Format NVM command. Bits 7:3 are reserved. Bit 2 indicates whether cryptographic erase is supported as part of the secure erase functionality. If set to '1', then cryptographic erase is supported. If cleared to '0', then cryptographic erase is not supported. Bit 1 indicates whether secure erase functionality applies to all namespaces in an NVM subsystem or is specific to a particular namespace. If set to '1', then any secure erase performed as part of a format operation results in a secure erase of all namespaces in the NVM subsystem. If cleared to '0', then any secure erase performed as part of a format results in a secure erase of the particular namespace specified. Bit 0 indicates whether the format operation (excluding secure erase) applies to all namespaces in an NVM subsystem or is specific to a particular namespace. If set to '1', then all namespaces in an NVM subsystem shall be configured with the same attributes and a format (excluding secure erase) of any namespace results in a format of all namespaces in an NVM subsystem. If cleared to '0', then the controller supports format on a per namespace basis.
525	M	**Volatile Write Cache (VWC):** This field indicates attributes related to the presence of a volatile write cache in the controller. Bits 7:1 are reserved. Bit 0 if set to '1' indicates that a volatile write cache is present. If cleared to '0', a volatile write cache is not present. If a volatile write cache is present, then the host controls whether the volatile write cache is enabled with Set Features specifying the Volatile Write Cache feature identifier (refer to 5.21.1.6). The Flush command (refer to section 6.8) is used to request that the contents of a volatile write cache be made non-volatile.
527:526	M	**Atomic Write Unit Normal (AWUN):** This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during normal operation. This field is specified in logical blocks and is a 0's based value. If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUN field in the Identify Namespace data structure. Refer to section 6.4. If a write command is submitted with size less than or equal to the AWUN value, the host is guaranteed that the write command is atomic to the NVM with respect to other read or write commands. If a write command is submitted with size greater than the AWUN value, then there is no guarantee of command atomicity. AWUN does not have any applicability to write errors caused by power failure (refer to Atomic Write Unit Power Fail). A value of FFFFh indicates all commands are atomic as this is the largest command size. It is recommended that implementations support a minimum of 128KB (appropriately scaled based on LBA size).

Bytes	O/M [1]	Description
529:528	M	**Atomic Write Unit Power Fail (AWUPF):** This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during a power fail or error condition. If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUPF field in the Identify Namespace data structure. Refer to section 6.4. This field is specified in logical blocks and is a 0's based value. The AWUPF value shall be less than or equal to the AWUN value. If a write command is submitted with size less than or equal to the AWUPF value, the host is guaranteed that the write is atomic to the NVM with respect to other read or write commands. If a write command is submitted that is greater than this size, there is no guarantee of command atomicity. If the write size is less than or equal to the AWUPF value and the write command fails, then subsequent read commands for the associated logical blocks shall return data from the previous successful write command. If a write command is submitted with size greater than the AWUPF value, then there is no guarantee of data returned on subsequent reads of the associated logical blocks.
530	M	**NVM Vendor Specific Command Configuration (NVSCC):** This field indicates the configuration settings for NVM Vendor Specific command handling. Refer to section 8.7. Bits 7:1 are reserved. Bit 0 if set to '1' indicates that all NVM Vendor Specific Commands use the format defined in Figure 12. If cleared to '0' indicates that the format of all NVM Vendor Specific Commands are vendor specific.
531	M	Reserved
533:532	O	**Atomic Compare & Write Unit (ACWU):** This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format for a Compare and Write fused operation. If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NACWU field in the Identify Namespace data structure. Refer to section 6.4. This field shall be supported if the Compare and Write fused command is supported. This field is specified in logical blocks and is a 0's based value. If a Compare and Write is submitted that requests a transfer size larger than this value, then the controller may fail the command with a status code of Invalid Field in Command. If Compare and Write is not a supported fused command, then this field shall be 0h.
535:534	M	Reserved

<table>
<tr><th>Bytes</th><th>O/M [1]</th><th>Description</th></tr>
<tr><td>539:536</td><td>O</td><td>SGL Support (SGLS): This field indicates if SGLs are supported for the NVM Command Set and the particular SGL types supported. Refer to section 4.4.

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31:21</td><td>Reserved</td></tr>
<tr><td>20</td><td>If set to ‘1’, then the controller supports the Address field in SGL Data Block, SGL Segment, and SGL Last Segment descriptor types specifying an offset. If cleared to ‘0’ then the Address field specifying an offset is not supported.</td></tr>
<tr><td>19</td><td>If set to ‘1’, then use of a Metadata Pointer (MPTR) that contains an address of an SGL segment containing exactly one SGL Descriptor that is Qword aligned is supported. If cleared to ‘0’, then use of a MPTR containing an SGL Descriptor is not supported.</td></tr>
<tr><td>18</td><td>If set to ‘1’, then the controller supports commands that contain a data or metadata SGL of a length larger than the amount of data to be transferred. If cleared to ‘0’, then the SGL length shall be equal to the amount of data to be transferred.</td></tr>
<tr><td>17</td><td>If set to ‘1’, then use of a byte aligned contiguous physical buffer of metadata (the Metadata Pointer field in Figure 11) is supported. If cleared to ‘0’, then use of a byte aligned contiguous physical buffer of metadata is not supported.</td></tr>
<tr><td>16</td><td>If set to ‘1’, then the SGL Bit Bucket descriptor is supported. If cleared to ‘0’, then the SGL Bit Bucket descriptor is not supported.</td></tr>
<tr><td>15:03</td><td>Reserved</td></tr>
<tr><td>02</td><td>If set to ‘1’, then the controller supports the Keyed SGL Data Block descriptor. If cleared to ‘0’, then the controller does not support the Keyed SGL Data Block descriptor.</td></tr>
<tr><td>01:00</td><td>This field is used to determine the SGL support for the NVM Command Set. Valid values are shown in the table below.

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>SGLs are not supported.</td></tr>
<tr><td>01b</td><td>SGLs are supported. There is no alignment nor granularity requirement for Data Blocks.</td></tr>
<tr><td>10b</td><td>SGLs are supported. There is a Dword alignment and granularity requirement for Data Blocks (refer to section 4.4).</td></tr>
<tr><td>11b</td><td>Reserved</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>767:540</td><td></td><td>Reserved</td></tr>
<tr><td>1023:768</td><td>M</td><td>NVM Subsystem NVMe Qualified Name (SUBNQN): This field specifies the NVM Subsystem NVMe Qualified Name as a UTF-8 null-terminated string. Refer to section 7.9 for the definition of NVMe Qualified Name.

Support for this field is mandatory if the controller supports revision 1.2.1 or later as indicated in the Version register (refer to section 3.1.2).</td></tr>
<tr><td>1791:1024</td><td></td><td>Reserved</td></tr>
<tr><td>2047:1792</td><td></td><td>Refer to the NVMe over Fabrics specification.</td></tr>
<tr><td colspan="3">Power State Descriptors</td></tr>
<tr><td>2079:2048</td><td>M</td><td>Power State 0 Descriptor (PSD0): This field indicates the characteristics of power state 0. The format of this field is defined in Figure 113.</td></tr>
<tr><td>2111:2080</td><td>O</td><td>Power State 1 Descriptor (PSD1): This field indicates the characteristics of power state 1. The format of this field is defined in Figure 113.</td></tr>
<tr><td>2143:2112</td><td>O</td><td>Power State 2 Descriptor (PSD2): This field indicates the characteristics of power state 2. The format of this field is defined in Figure 113.</td></tr>
<tr><td>2175:2144</td><td>O</td><td>Power State 3 Descriptor (PSD3): This field indicates the characteristics of power state 3. The format of this field is defined in Figure 113.</td></tr>
</table>

Bytes	O/M [1]	Description
2207:2176	O	**Power State 4 Descriptor (PSD4):** This field indicates the characteristics of power state 4. The format of this field is defined in Figure 113.
2239:2208	O	**Power State 5 Descriptor (PSD5):** This field indicates the characteristics of power state 5. The format of this field is defined in Figure 113..
2271:2240	O	**Power State 6 Descriptor (PSD6):** This field indicates the characteristics of power state 6. The format of this field is defined in Figure 113.
2303:2272	O	**Power State 7 Descriptor (PSD7):** This field indicates the characteristics of power state 7. The format of this field is defined in Figure 113.
2335:2304	O	**Power State 8 Descriptor (PSD8):** This field indicates the characteristics of power state 8. The format of this field is defined in Figure 113.
2367:2336	O	**Power State 9 Descriptor (PSD9):** This field indicates the characteristics of power state 9. The format of this field is defined in Figure 113.
2399:2368	O	**Power State 10 Descriptor (PSD10):** This field indicates the characteristics of power state 10. The format of this field is defined in Figure 113.
2431:2400	O	**Power State 11 Descriptor (PSD11):** This field indicates the characteristics of power state 11. The format of this field is defined in Figure 113.
2463:2432	O	**Power State 12 Descriptor (PSD12):** This field indicates the characteristics of power state 12. The format of this field is defined in Figure 113.
2495:2464	O	**Power State 13 Descriptor (PSD13):** This field indicates the characteristics of power state 13. The format of this field is defined in Figure 113.
2527:2496	O	**Power State 14 Descriptor (PSD14):** This field indicates the characteristics of power state 14. The format of this field is defined in Figure 113.
2559:2528	O	**Power State 15 Descriptor (PSD15):** This field indicates the characteristics of power state 15. The format of this field is defined in Figure 113.
2591:2560	O	**Power State 16 Descriptor (PSD16):** This field indicates the characteristics of power state 16. The format of this field is defined in Figure 113.
2623:2592	O	**Power State 17 Descriptor (PSD17):** This field indicates the characteristics of power state 17. The format of this field is defined in Figure 113.
2655:2624	O	**Power State 18 Descriptor (PSD18):** This field indicates the characteristics of power state 18. The format of this field is defined in Figure 113.
2687:2656	O	**Power State 19 Descriptor (PSD19):** This field indicates the characteristics of power state 19. The format of this field is defined in Figure 113.
2719:2688	O	**Power State 20 Descriptor (PSD20):** This field indicates the characteristics of power state 20. The format of this field is defined in Figure 113.
2751:2720	O	**Power State 21 Descriptor (PSD21):** This field indicates the characteristics of power state 21. The format of this field is defined in Figure 113.
2783:2752	O	**Power State 22 Descriptor (PSD22):** This field indicates the characteristics of power state 22. The format of this field is defined in Figure 113.
2815:2784	O	**Power State 23 Descriptor (PSD23):** This field indicates the characteristics of power state 23. The format of this field is defined in Figure 113.
2847:2816	O	**Power State 24 Descriptor (PSD24):** This field indicates the characteristics of power state 24. The format of this field is defined in Figure 113.
2879:2848	O	**Power State 25 Descriptor (PSD25):** This field indicates the characteristics of power state 25. The format of this field is defined in Figure 113.
2911:2880	O	**Power State 26 Descriptor (PSD26):** This field indicates the characteristics of power state 26. The format of this field is defined in Figure 113.
2943:2912	O	**Power State 27 Descriptor (PSD27):** This field indicates the characteristics of power state 27. The format of this field is defined in Figure 113.
2975:2944	O	**Power State 28 Descriptor (PSD28):** This field indicates the characteristics of power state 28. The format of this field is defined in Figure 113.
3007:2976	O	**Power State 29 Descriptor (PSD29):** This field indicates the characteristics of power state 29. The format of this field is defined in Figure 113.
3039:3008	O	**Power State 30 Descriptor (PSD30):** This field indicates the characteristics of power state 30. The format of this field is defined in Figure 113.
3071:3040	O	**Power State 31 Descriptor (PSD31):** This field indicates the characteristics of power state 31. The format of this field is defined in Figure 113.
Vendor Specific		
4095:3072	O	Vendor Specific.

Bytes	O/M [1]	Description
NOTES:		
1 O/M definition: O = Optional, M = Mandatory.		

Figure 113 defines the power state descriptor that describes the attributes of each power state. For more information on how the power state descriptor fields are used, refer to section 8.4 on power management.

Figure 113: Identify – Power State Descriptor Data Structure

Bits	Description
255:184	Reserved
183:182	**Active Power Scale (APS):** This field indicates the scale for the Active Power field. If an Active Power Workload is reported for a power state, then the Active Power Scale shall also be reported for that power state. **Value** / **Definition** 00b – Not reported for this power state 01b – 0.0001 W 10b – 0.01 W 11b – Reserved
181:179	Reserved
178:176	**Active Power Workload (APW):** This field indicates the workload used to calculate maximum power for this power state. Refer to section 8.4.3 for more details on each of the defined workloads. This field shall not be "No Workload" unless ACTP is 0000h.
175:160	**Active Power (ACTP):** This field indicates the largest average power consumed by the NVM subsystem over a 10 second period in this power state with the workload indicated in the Active Power Workload field. The power in Watts is equal to the value in this field multiplied by the scale indicated in the Active Power Scale field. A value of 0000h indicates Active Power is not reported.
159:152	Reserved
151:150	**Idle Power Scale (IPS):** This field indicates the scale for the Idle Power field. **Value** / **Definition** 00b – Not reported for this power state 01b – 0.0001 W 10b – 0.01 W 11b – Reserved
149:144	Reserved
143:128	**Idle Power (IDLP):** This field indicates the typical power consumed by the NVM subsystem over 30 seconds in this power state when idle (i.e., there are no pending commands, register accesses, background processes, nor device self-test operations). The measurement starts after the NVM subsystem has been idle for 10 seconds. The power in Watts is equal to the value in this field multiplied by the scale indicated in the Idle Power Scale field. A value of 0000h indicates Idle Power is not reported.
127:125	Reserved
124:120	**Relative Write Latency (RWL):** This field indicates the relative write latency associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means lower write latency.
119:117	Reserved
116:112	**Relative Write Throughput (RWT):** This field indicates the relative write throughput associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means higher write throughput.
111:109	Reserved
108:104	**Relative Read Latency (RRL):** This field indicates the relative read latency associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means lower read latency.

Bits	Description
103:101	Reserved
100:96	**Relative Read Throughput (RRT):** This field indicates the relative read throughput associated with this power state. The value in this field shall be less than the number of supported power states (e.g., if the controller supports 16 power states, then valid values are 0 through 15). A lower value means higher read throughput.
95:64	**Exit Latency (EXLAT):** This field indicates the maximum exit latency in microseconds associated with exiting this power state. A value of 0h indicates Exit Latency is not reported.
63:32	**Entry Latency (ENLAT):** This field indicates the maximum entry latency in microseconds associated with entering this power state. A value of 0h indicates Entry Latency is not reported.
31:26	Reserved
25	**Non-Operational State (NOPS):** This field indicates whether the controller processes I/O commands in this power state. If this field is cleared to '0', then the controller processes I/O commands in this power state. If this field is set to '1', then the controller does not process I/O commands in this power state. Refer to section 8.4.1.
24	**Max Power Scale (MXPS):** This field indicates the scale for the Maximum Power field. If this field is cleared to '0', then the scale of the Maximum Power field is in 0.01 Watts. If this field is set to '1', then the scale of the Maximum Power field is in 0.0001 Watts.
23:16	Reserved
15:00	**Maximum Power (MP):** This field indicates the maximum power consumed by the NVM subsystem in this power state. The power in Watts is equal to the value in this field multiplied by the scale specified in the Max Power Scale field. A value of 0h indicates Maximum Power is not reported. Refer to section 8.4.

5.15.4 Active Namespace ID list (CNS 02h)

A list of 1024 namespace IDs is returned to the host containing active NSIDs in increasing order that are greater than the value specified in the Namespace Identifier (CDW1.NSID) field of the command. The controller should abort the command with status code Invalid Namespace or Format if CDW1.NSID is set to FFFFFFFEh or FFFFFFFFh. ~~Note that~~ The CDW1.NSID field may be cleared to 0h to retrieve a Namespace List including the namespace starting with NSID of 1h. The data structure returned is a Namespace List (refer to section 4.8).

5.15.5 Namespace Identification Descriptor list (CNS 03h)

A list of Namespace Identification Descriptor structures (refer to Figure 116) is returned to the host for the namespace specified in the Namespace Identifier (CDW1.NSID) field if it is an active NSID.

The controller may return any number of variable length Namespace Identification Descriptor structures that fit into the 4096 byte Identify payload. All remaining bytes after the namespace identification descriptor structures should be cleared to 0h, and the host shall interpret a Namespace Identifier Descriptor Length (NIDL) value of 0h as the end of the list. If, while processing these descriptors, the host detects a descriptor type that it does not recognize, then it should skip the unrecognized descriptor type and continue parsing the structure.

A controller shall not return multiple descriptors with the same Namespace Identifier Type (NIDT). A controller shall return at least one descriptor identifying the namespace.

Figure 116: Identify – Namespace Identification Descriptor

<table>
<tr><th>Bytes</th><th>Description</th></tr>
<tr><td>0</td><td>Namespace Identifier Type (NIDT): This field indicates the data type contained in the Namespace Identifier field and the length for that type as defined in the following table.
<table>
<tr><th>Value</th><th>Length (NIDL)</th><th>Definition</th></tr>
<tr><td>0h</td><td></td><td>Reserved</td></tr>
<tr><td>1h</td><td>8h</td><td>IEEE Extended Unique Identifier: The NID field contains a copy of the EUI64 field in the Identify Namespace data structure (refer to Figure 114). If the EUI64 field of the Identify Namespace data structure is not supported (i.e., EUI64 field is set to zero), the controller shall not report a Namespace Identification Descriptor with a value of type 1h.</td></tr>
<tr><td>2h</td><td>10h</td><td>Namespace Globally Unique Identifier: The NID field contains a copy of the NGUID field in the Identify Namespace data structure (refer to Figure 114). If the NGUID field of the Identify Namespace data structure is not supported (i.e., the NGUID field is set to zero), the controller shall not report a Namespace Identification Descriptor with a value of type 2h.</td></tr>
<tr><td>3h</td><td>10h</td><td>Namespace UUID: The NID field contains a 128-bit Universally Unique Identifier (UUID) as specified in RFC4122.

If the namespace does not support an IEEE Extended Unique Identifier (i.e., EUI64 field is set to zero) and does not support a Namespace Globally Unique Identifier (i.e., the NGUID field is set to zero), then the namespace shall report a Namespace Identification Descriptor with a value of type 3h.</td></tr>
<tr><td>4h - FFh</td><td></td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>1</td><td>Namespace Identifier Length (NIDL): This field contains the length in bytes of the Namespace Identifier (NID) field. The total length of the Namespace Identification Descriptor in bytes is the value in this field plus four. If this field is set to 0h it indicates the end of the Namespace Identifier Descriptor list.</td></tr>
<tr><td>3:2</td><td>Reserved</td></tr>
<tr><td>(NIDL+3):4</td><td>Namespace Identifier (NID): This field contains a value that is globally unique and assigned to the namespace when the namespace is created. This field remains fixed throughout the life of the namespace and is preserved across namespace and controller operations (e.g., controller reset, namespace format, etc.). The type of the value is specified by the Namespace Identifier Type (NIDT) field, and the size is specified by the Namespace Identifier Length (NIDL) field.</td></tr>
</table>

5.15.6 Allocated Namespace ID list (CNS 10h)

A list of up to 1024 namespace IDs is returned to the host containing allocated NSIDs in increasing order that are ~~with a namespace identifier~~ greater than the value specified in the Namespace Identifier (CDW1.NSID) field of the Identify command.

The controller should abort the command with status code Invalid Namespace or Format if CDW1.NSID is set to FFFFFFFEh or FFFFFFFFh. ~~Note that~~ The CDW1.NSID may be cleared to 0h to retrieve a Namespace List including the namespace starting with NSID of 1h. The data structure returned is a Namespace List (refer to section 4.8).

5.15.7 Identify Allocated Namespace data structure (CNS 11h)

The Identify Namespace data structure (refer to Figure 114) is returned to the host for the namespace specified in the Namespace Identifier (CDW1.NSID) field if it is an allocated NSID. If the specified namespace is an unallocated NSID then the controller returns a zero filled data structure.

If the specified namespace is an invalid NSID then the controller shall fail the command with a status code of Invalid Namespace or Format. If CDW1.NSID is set to FFFFFFFFh then the controller should fail the command with a status code of Invalid Namespace or Format.

5.15.8 Namespace Attached Controller list (CNS 12h)

A Controller List of up to 2047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers that are attached to the namespace specified in the Namespace Identifier (CDW1.NSID) field.

5.15.9 Controller list (CNS 13h)

A Controller List of up to 2047 controller identifiers is returned containing a controller identifier greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field. The list contains controller identifiers in the NVM subsystem that may or may not be attached to namespace(s).

5.15.10 Primary Controller Capabilities data structure (CNS 14h)

The Primary Controller Capabilities Structure (refer to Figure 110) is returned to the host for the primary controller specified.

Figure 110: Identify – Primary Controller Capabilities Structure

Bytes	Description
1:0	**Controller Identifier (CNTLID):** This field indicates the Controller Identifier of the primary controller.
3:2	**Port Identifier (PORTID):** This field indicates the Port Identifier of the NVM subsystem port associated with the primary controller. The Port Identifier for a PCI Express Port is the same as the Port Number field in Link Capabilities Register in the PCI Express Capability structure (refer to section 2.5.6).
4	**Controller Resource Types (CRT):** This field indicates the controller resources types supported. If a controller resource type is supported, then it shall be supported for the primary controller and all associated secondary controllers. Bits 7:2 are reserved. Bit 1 if set to '1' then VI Resources are supported. Bit 1 if cleared to '0' then VI Resources are not supported. Refer to section 8.5.2 Bit 0 if set to '1' then VQ Resources are supported. Bit 0 if cleared to '0' then VQ Resources are not supported. Refer to section 8.5.1
31:5	Reserved
35:32	**VQ Resources Flexible Total (VQFRT):** This field indicates the total number of VQ Flexible Resources for the primary and its secondary controllers.
39:36	**VQ Resources Flexible Assigned (VQRFA):** This field indicates the total number of VQ Flexible Resources Assigned to the associated secondary controllers.
41:40	**VQ Resources Flexible Allocated to Primary (VQRFAP):** This field indicates the total number of VQ Flexible Resources currently allocated to the primary controller. This value may change after a Controller Level Reset other than a Controller Reset (i.e., CC.EN transitions from '1' to '0') if a new value was set using the Virtualization Management command. The default value of this field is implementation specific.
43:42	**VQ Resources Private Total (VQPRT):** This field indicates the total number of VQ Private Resources for the primary controller.
45:44	**VQ Resources Flexible Secondary Maximum (VQFRSM):** This field indicates the maximum number of VQ Flexible Resources that may be assigned to a secondary controller.
47:46	**VQ Flexible Resource Preferred Granularity (VQGRAN):** This field indicates the preferred granularity of assigning and removing VQ Flexible Resources. Assigning and removing VQ Resources in this granularity minimizes any wasted internal implementation resources.
63:48	Reserved
67:64	**VI Resources Flexible Total (VIFRT):** This field indicates the total number of VI Flexible Resources for the primary and its secondary controllers.
71:68	**VI Resources Flexible Assigned (VIRFA):** This field indicates the total number of VI Flexible Resources Assigned to the associated secondary controllers.

Bytes	Description
73:72	**VI Resources Flexible Allocated to Primary (VIRFAP):** This field indicates the total number of VI Flexible Resources currently allocated to the primary controller. This value may change after a Controller Level Reset other than a Controller Reset (i.e., CC.EN transitions from '1' to '0') if a new value was set using the Virtualization Management command. The default value of this field is implementation specific.
75:74	**VI Resources Private Total (VIPRT):** This field indicates the total number of VI Private Resources for the primary controller.
77:76	**VI Resources Flexible Secondary Maximum (VIFRSM):** This field indicates the maximum number of VI Flexible Resources that may be assigned to a secondary controller.
79:78	**VI Flexible Resource Preferred Granularity (VIGRAN):** This field indicates the preferred granularity of assigning and removing VI Flexible Resources. Assigning and removing VI Resources in this granularity minimizes any wasted internal implementation resources.
4095:80	Reserved

5.15.11 Secondary Controller list (CNS 15h)

A Secondary Controller List (refer to Figure 111) is returned to the host for up to 127 secondary controllers associated with the primary controller ~~issuing~~ processing this command. The list contains entries for controller identifiers greater than or equal to the value specified in the Controller Identifier (CDW10.CNTID) field.

~~Figure defines a Secondary Controller List.~~ All secondary controllers are represented, including those that are in an Offline state due to SR-IOV configuration settings (e.g., VF Enable is cleared to 0h or NumVFs specifies a value that does not enable the associated secondary controller).

Figure 111: Secondary Controller List

Bytes	Description
0	**Number of Identifiers:** This field indicates the number of Secondary Controller Entries in the list. There are up to 127 entries in the list. A value of 0 indicates there are no entries in the list.
31:1	Reserved
63:32	**SC Entry 0:** This field contains the first Secondary Controller Entry in the list, if present.
95:64	**SC Entry 1:** This field contains the second Secondary Controller Entry in the list, if present.
…	…
(N*32+63): (N*32+32)	**SC Entry N:** This field contains the N+1 Secondary Controller Entry in the list, if present.

Figure 112: Secondary Controller Entry

Bytes	Description
1:0	**Secondary Controller Identifier (SCID):** This field indicates the Controller Identifier of the secondary controller described by this entry.
3:2	**Primary Controller Identifier (PCID): T**his field indicates the Controller Identifier of the associated primary controller.
4	**Secondary Controller State (SCS):** This field indicates the state of the secondary controller. Bits 7:1 are reserved. Bit 0 if set to '1' then the controller is in the Online state. Bit 0 if cleared to '0' then the controller is in the Offline state.
7:5	Reserved
9:8	**Virtual Function Number (VFN):** If the secondary controller is an SR-IOV VF, this field indicates its VF Number, where VF Number > 0, and VF Number is no larger than the total number of VFs indicated by the TotalVFs register (refer to Single Root I/O Virtualization and Sharing Specification) in the PF's SR-IOV Extended Capability structure. If the secondary controller is not an SR-IOV VF, then this field is cleared to zero.
11:10	**Number of VQ Flexible Resources Assigned (NVQ):** This field indicates the number of VQ Flexible Resources currently assigned to the indicated secondary controller.
13:12	**Number of VI Flexible Resources Assigned (NVI):** This field indicates the number of VI Flexible Resources currently assigned to the indicated secondary controller.

Bytes	Description
31:14	Reserved

5.15.12 Command Completion

Upon completion of the Identify command, the controller posts a completion queue entry to the Admin Completion Queue.

Modify a portion of section 5.19 (Namespace Attachment command) and section 5.20 (Namespace Management command) as show below:

5.19 Namespace Attachment command

The Namespace Attachment command is used to attach and detach controllers from a namespace. The attach and detach operations are persistent across all reset events.

If the Namespace Attachment command is supported, then the Namespace Management command (refer to section 5.20) shall also be supported.

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

…

5.20 Namespace Management command

The Namespace Management command is used to manage namespaces (refer to section 8.12), including create and delete operations.

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

If the Namespace Management command is supported, then the Namespace Attachment command (refer to section 5.19) shall also be supported.

Host software uses the Namespace Attachment command to attach or detach a namespace **…**

…

The data structure used for the create operation is defined in Figure 129 and has the same format as the Identify Namespace data structure defined in Figure 114. After successful completion of a Namespace Management command with the create operation, the namespace is formatted with the specified attributes. The fields that host software may specify in the create operation are defined in Figure 126. Fields that are reserved ~~shall be~~ are cleared to 0h by host software. There is no data structure transferred for the delete operation.

…

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

The Namespace Identifier (CDW1.NSID) field is used as follows for create and delete operations:

- Create: The CDW1.NSID field is reserved for this operation; host software ~~shall set~~ clears this field to a value of 0h. The controller shall select an available Namespace Identifier to use for the operation.
- Delete: This field specifies the previously created namespace to delete in this operation. Specifying a value of FFFFFFFFh is used to delete all namespaces in the NVM subsystem. If the value of FFFFFFFFh is specified and there are zero valid namespaces, the command completes successfully.

…

Modify a portion of section 5.21 (Set Features command) as shown below:

5.21 Set Features command

…

5.21.1 Feature Specific Information

…

5.21.1.3 LBA Range Type (Feature Identifier 03h), (Optional)

This feature indicates the type and attributes of LBA ranges **…**

For a Get Features command, the controller shall clear to zero all unused entries in the LBA Range Type data structure. For a Set Features command, the controller shall ignore all unused entries in the LBA Range Type data structure.

If the size of the namespace or the LBA format of the namespace changes, the specified LBA ranges may not represent the expected locations in the NVM. After such a change, the host should ensure the intended LBAs are specified.

The default value for this feature should clear **…**

…

5.21.1.9 Interrupt Vector Configuration (Feature Identifier 09h)

…

Prior to issuing a Set Features command that specifies this Feature, the host shall configure the specified Interrupt Vector with an ~~valid~~ I/O Completion Queue (refer to section 5.3). If the ~~I/O Completion Queue or~~ specified Interrupt Vector ~~specified~~ is invalid, or not associated with an existing I/O Completion Queue (refer to Figure 53), then the controller should return an error of Invalid Field in Command.

Figure 141: Interrupt Vector Configuration – Command Dword 11

Bit	Description
31:17	Reserved
16	**Coalescing Disable (CD):** If set to '1', then any interrupt coalescing settings shall not be applied for this interrupt vector. If cleared to '0', then interrupt coalescing settings apply for this interrupt vector.
15:00	**Interrupt Vector (IV):** This field specifies the interrupt vector for which the configuration settings are applied.

…

5.21.1.12 Autonomous Power State Transition (Feature Identifier 0Ch), (Optional)

This feature configures the settings for autonomous power state transitions, refer to section 8.4.2.

The Autonomous Power State Transition uses Command Dword 11 and specifies the attribute information in the data structure indicated in Figure 149 and the Autonomous Power State Transition data structure consisting of 32 of the entries defined in Figure 150.

If a Get Features command is issued for this Feature, the attributes specified in Figure 149 are returned in Dword 0 of the completion queue entry and the Autonomous Power State Transition data structure, whose entry structure is defined in Figure 150, is returned in the data buffer for that command.

Figure 144: Autonomous Power State Transition – Command Dword 11

Bit	Description
31:01	Reserved
00	**Autonomous Power State Transition Enable (APSTE):** This field specifies whether autonomous power state transition is enabled. If this field is set to '1', then autonomous power state transitions are enabled. If this field is cleared to '0', then autonomous power state transitions are disabled. This field is cleared to '0' by default.

Each entry in the Autonomous Power State Transition data structure is defined in Figure 150. Each entry is 64 bits in size. There is an entry for each of the allowable 32 power states. For power states that are not supported,

the unused Autonomous Power State Transition data structure entries shall be cleared to all zeroes. The entries begin with power state 0 and then increase sequentially (i.e., power state 0 is described in bytes 7:0, power state 1 is described in bytes 15:8, etc.). The data structure is 256 bytes in size and shall be physically contiguous.

Figure 145: Autonomous Power State Transition – Data Structure Entry

Bit	Description
63:32	Reserved
31:08	**Idle Time Prior to Transition (ITPT):** This field specifies the amount of idle time that occurs in this power state prior to transitioning to the Idle Transition Power State. The time is specified in milliseconds. A value of 0h disables the autonomous power state transition feature for this power state.
07:03	**Idle Transition Power State (ITPS):** This field specifies the power state for the controller to autonomously transition to after there is a continuous period of idle time in the current power state that exceeds time specified in the Idle Time Prior to Transition field. The field specified is required to be a non-operational state as described in ~~Figure 113~~ Figure 113. This field should not specify a power state with higher reported idle power than the current power state.
02:00	Reserved

The Autonomous Power State Transition feature may interact with the Non-Operational Power State Config feature (refer to section 5.21.1.17). Figure 145.a shows these interactions.

Figure 145.a: Interactions between APSTE and NOPPME

APSTE[1]	NOPPME[2]	Non-operational power state entry	Background operations during non-operational power states
1	1	Entered by host request[3] or by ITPT idle timer[4]	Allowed
0	1	Entered by host request[3]	Allowed
1	0	Entered by host request[3] or by ITPT idle timer[4]	Not allowed
0	0	Entered by host request[3]	Not allowed

1. Defined in Figure 144.
2. Defined in Figure 159.
3. Refer to section 5.21.1.1.
4. Refer to Figure 145.

Modify a portion of section 5.21.1.17 (Non-Operational Power State Config) as shown below:

5.21.1.17 Non-Operational Power State Config (Feature Identifier 11h), (Optional)

This Feature configures non-operational power state settings for the controller. The settings are specified in Command Dword 11.

If a Get Features command is submitted for this Feature, the values in Figure 157 are returned in Dword 0 of the completion queue entry for that command.

Figure 159: Non-Operational Power State Config – Command Dword 11

Bit	Description
31:01	Reserved
00	**Non-Operational Power State Permissive Mode Enable (NOPPME):** If NOPPME is set to '1' then the controller may temporarily exceed the power limits of any non-operational power state, up to the limits of the last operational power state, to run controller initiated background operations in that state (i.e., Non-Operational Power State Permissive Mode is enabled). If NOPPME is cleared to '0', then the controller shall not exceed the limits of any non-operational state while running controller initiated background operations in that state (i.e., Non-Operational Power State Permissive Mode is disabled). If Non-Operational Power State Permissive Mode is disabled, then: a) thermal management that requires power (e.g., cooling fans) may be disabled; and b) performance after resuming from the non-operational power state may be degraded until background activity that was not allowed while in that non-operational power state has completed. If the host attempts to set this field to '1' and the controller does not support Non-Operational Power State Permissive Mode as indicated in the Controller Attributes field of Identify Controller, then the command fails with a status of Invalid Field in Command.

The Non-Operational Power State Config feature may interact with the Autonomous Power State Transition feature (refer to section 5.21.1.12). Figure 144.a shows these interactions.

…

Modify a portion of 5.21.1.19 (Host Identifier) as follows:

5.21.1.19 Host Identifier (Feature Identifier 81h), (Optional[1])

…

Figure 161: Host Identifier – Command Dword 11

Bit	Description
31:01	Reserved

[1] Mandatory if reservations are supported as indicated in the Identify Controller data structure.

Bit	Description
00	**Enable Extended Host Identifier (EXHID):** If set to '1', then the host is ~~using~~ requesting the use of an extended 128-bit Host Identifier. If cleared to '0', then the host is ~~using~~ requesting the use of a 64-bit Host Identifier. NVMe over Fabrics implementations shall use an extended 128-bit Host Identifier. If the controller does not support a 128-bit Host Identifier as indicated in the Controller Attributes field in the Identify Controller data structure and the host sets this bit to '1', then a status value of Invalid Field in Command shall be returned. If the controller does not support a 64-bit Host Identifier (e.g., the device is an NVMe over Fabrics device) and the host clears this bit to '0', then a status value of Invalid Field in Command shall be returned. If the NVM subsystem supports a 64-bit Host Identifier, supports a 128-bit Host Identifier and detects that another controller in the NVM subsystem is already using a non-zero Host Identifier of a different size than ~~specified~~ the size requested in this command, then a status of Host Identifier Inconsistent Format shall be returned.

…

5.21.1.19.2 NVMe over Fabrics

The Host Identifier is a mandatory feature in NVMe over Fabrics. The Host Identifier shall be an extended 128-bit Host Identifier. The Host Identifier shall be set to a non-zero value in the Fabrics Connect command. The Host Identifier shall not be modified. A Set Features command specifying the Host Identifier Feature shall be aborted with a status of Command Sequence Error. A Get Features command specifying the Host Identifier Feature shall return the value set in the Fabrics Connect command. ~~A Get Features command specifying a 64-bit Host Identifier (EXHID cleared to '0') shall be aborted with a status of Invalid Field in Command.~~

Modify a portion of 5.22 (Virtualization Management command) as follows:

5.22 Virtualization Management command

The Virtualization Management command is supported by primary controllers that support the Virtualization Enhancements capability. This command is used for several functions:

- Modifying Flexible Resource allocation for the primary controller;
- Assigning Flexible Resources for secondary controllers; and
- Setting the Online and Offline state for secondary controllers.

Refer to section 8.5 for more on the Virtualization Enhancements capability and the Virtualization Management command.

The Virtualization Management command uses the Command Dword 10 and Command Dword 11 fields. All other command specific fields are reserved.

If the action requested specifies a range of controller resources that:

a) does not exist~~,~~;
b) is a Private Resource (e.g., VQ resources are requested when VQ resources are not supported, VI resources are requested when VI resources are not supported)~~,~~; or
c) is currently in use (e.g., the number of Controller Resources (NR) is greater than the number of remaining available flexible resources),

then an error of Invalid Resource Identifier is returned.

…

5.22.1 Command Completion

Command specific status values associated with the Virtualization management command are defined in Figure 168.

Figure 168: Virtualization Management – Command Specific Status Values

Value	Description
1Fh	**Invalid Controller Identifier:** An invalid Controller Identifier was specified.
20h	**Invalid Secondary Controller State:** The action requested for the secondary controller is invalid based on the current state of the secondary controller and its primary controller.
21h	**Invalid Number of Controller Resources:** The specified number of Flexible Resources is invalid (e.g., the Number of Controller Resources (NR) is greater than VQ Resources Flexible Total (VQFRT) (refer to Figure 110), the Number of Controller Resources (NR) is greater than VQ Resources Flexible Secondary Maximum (VQFRSM) (refer to Figure 110)).
22h	**Invalid Resource Identifier:** At least one of the specified resource identifiers was invalid (e.g., the Number of Controller Resources (NR) is greater than the number of remaining available flexible resources).

Modify a portion of section 5.23 (Format NVM command) as shown below (these changes are based on changes published in ECN-004):

5.23 Format NVM command – NVM Command Set Specific

…

The Format NVM command shall fail if the controller is in an invalid security state (refer to the appropriate security specification, e.g., TCG Storage Interface Interactions Specification). The Format NVM command may fail if there are outstanding I/O commands to the namespace specified to be formatted. I/O commands for a namespace that has a Format NVM command in progress may fail.

For a Format command with an NSID field set to FFFFFFFFh that specifies secure erase:

a) if bit 1 is set to '1' in the FNA field (refer to Figure 109) and there are no namespaces in the NVM subsystem, then that Format command shall complete without error; and
b) if bit 1 is cleared to '0' in the FNA field and there are no attached namespaces, then that Format command shall complete without error.

For a Format command with an NSID field set to FFFFFFFFh that does not specify a secure erase:

a) if bit 0 is set to '1' in the FNA field and there are no namespaces in the NVM subsystem, then that Format command shall complete without error; and
b) if bit 0 is cleared to '0' in the FNA field and there are no attached namespaces, then that Format command shall complete without error.

…

Modify a portion of section 5.24 (Sanitize command) as shown below:

5.24 Sanitize command – NVM Command Set Specific

…

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

- Shall clear any outstanding Sanitize Operation Completed asynchronous event;
- Shall update the Sanitize Status log (refer to section 5.15.1.12.2);

- Shall abort any command (submitted or in progress) not allowed during a sanitize operation with a status of Sanitize In Progress (refer to section 8.15.1);
- ~~Should s~~Suspends autonomous power state management activities as described in section 8.4.2; and
- Shall release stream identifiers for any open streams.

…

Modify a portion of section 6 (NVM Command Set) as shown below:

6 NVM Command Set

…

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

Figure 184: Opcodes for NVM Commands

Opcode by Field			Combined Opcode[2]	O/M[1]	Command[3]	Reference Section
(07)	(06:02)	(01:00)				
Standard Command	Function	Data Transfer[5]				
0b	000 00b	00b	00h	M	Flush	6.8
0b	000 00b	01b	01h	M	Write	6.14
0b	000 00b	10b	02h	M	Read	6.9
0b	000 01b	00b	04h	O	Write Uncorrectable	6.15
0b	000 01b	01b	05h	O	Compare	6.6
0b	000 10b	00b	08h	O	Write Zeroes	6.16
0b	000 10b	01b	09h	O	Dataset Management	6.7
0b	000 11b	01b	0Dh	O[4]	Reservation Register	6.11
0b	000 11b	10b	0Eh	O[4]	Reservation Report	6.13
0b	001 00b	01b	11h	O[4]	Reservation Acquire	6.10
0b	001 01b	01b	15h	O[4]	Reservation Release	6.12
	Vendor Specific					
1b	na	NOTE 5	80h – FFh	O	Vendor specific	

NOTES:

1. O/M definition: O = Optional, M = Mandatory.
2. Opcodes not listed are reserved.
3. All NVM commands use the Namespace Identifier field (CDW1.NSID).
4. Mandatory if reservations are supported as indicated in the Identify Controller data structure.
5. Indicates the data transfer direction of the command. All options to the command shall transfer data as specified or transfer no data. All commands, including vendor specific commands, shall follow this convention: 00b = no data transfer; 01b = host to controller; 10b = controller to host; 11b = bidirectional.

6.1 Namespaces

6.1.1 Namespace Overview

A namespace is a collection of logical blocks whose logical block addresses range from 0 to the ~~capacity~~ size of the namespace – 1. A namespace ID (NSID) is an identifier used by a controller to provide access to a namespace.

…

6.1.6 NSID and Namespace Usage

If the Namespace Management capability is supported (refer to section 8.12 ~~the OACS field in Figure 109~~) then NSIDs shall be unique within the NVM subsystem (e.g., NSID of 3 shall refer to the same physical namespace regardless of the accessing controller). If the Namespace Management capability is not supported, then NSIDs:

…

Namespace IDs may change across power off conditions ~~or due to namespace management~~. However, it is recommended that namespace identifiers remain static across power off conditions in order to avoid issues with EFI or OSes. To determine if the same namespace has been encountered, the host may use the:

a) UUID field in the Namespace Identification Descriptor (refer to Figure 116), if present;
b) NGUID field in the Identify Namespace data (refer to Figure 114) or in the Namespace Identification Descriptor, if present; or
c) EUI64 field in the Identify Namespace data or in the Namespace Identification Descriptor, if present.

Bit 3 in the NSFEAT field (refer to Figure 114) indicates NGUID and EUI64 reuse characteristics.

The Namespace Size field in the Identify Namespace data structure **…**

…

A namespace may or may not have a relationship to a Submission Queue; this relationship is determined by the host software implementation. The controller shall support access to any ~~valid~~ active namespace from any I/O Submission Queue.

Modify a portion of section 6.10 (Reservation Acquire command) as shown below:

6.10 Reservation Acquire command

…

Figure 216: Reservation Acquire Data Structure

Bytes	~~O/M~~	Description
7:0	~~M~~	**Current Reservation Key (CRKEY):** The field specifies the current reservation key associated with the host.
15:8	~~M~~	**Preempt Reservation Key (PRKEY):** If the Reservation Acquire Action is set to 001b (i.e., Preempt) or 010b (i.e., Preempt and Abort), then this field specifies the reservation key to be unregistered from the namespace. For all other Reservation Acquire Action values, this field is reserved.

Modify a portion of section 6.11 (Reservation Register command) as shown below:

6.11 Reservation Register command

…

Figure 220: Reservation Register Data Structure

Bytes	~~O/M~~	Description
7:0	~~M~~	**Current Reservation Key (CRKEY):** If the Reservation Register Action is 001b (i.e., Unregister Reservation Key) or 010b (i.e., Replace Reservation Key), then this field contains the current reservation key associated with the host. For all other Reservation Register Action values, this field is reserved. The controller ignores the value of this field when the Ignore Existing Key (IEKEY) bit is set to '1'.
15:8	~~M~~	**New Reservation Key (NRKEY):** If the Reservation Register Action is 000b (i.e., Register Reservation Key) or 010b (i.e., Replace Reservation Key), then this field contains the new reservation key associated with the host. For all other Reservation Register Action values, this field is reserved.

Modify a portion of section 7.5 (Interrupts) as shown below:

7.5 Interrupts

…

7.5.1 Pin Based, Single MSI, and Multiple MSI Behavior

This is the mode of interrupt operation if any of the following conditions are met:

- Pin based interrupts are being used – MSI (MSICAP.MC.MSIE='0') and MSI-X are disabled;
- Single MSI is being used – MSI is enabled (MSICAP.MC.MSIE='1'), MSICAP.MC.MME=0h, and MSI-X is disabled; or
- Multiple MSI is being used – Multiple-message MSI is enabled (MSICAP.MC.MSIE='1'), MSICAP.MC.MME is set to a value between 001b and 101b inclusive, and MSI-X is disabled.

Within the controller there is an interrupt status register (IS) that is not visible to the host. In this mode, the IS register determines whether the PCI interrupt line shall be driven active or an MSI message shall be sent. Each bit in the IS register corresponds to an interrupt vector. The IS bit is set to '1' when the ~~AND of the~~ following conditions ~~is~~ are true:

- There is one or more unacknowledged completion queue entries in a Completion Queue that utilizes this interrupt vector;
- The Completion Queue(s) with unacknowledged completion queue entries has interrupts enabled in the "Create I/O Completion Queue" command; and
- The corresponding INTM bit exposed to the host is cleared to '0', indicating that the interrupt is not masked.

Editors note: Lists should use semicolon (";") separators rather than comma (",") as well as having an "and" or an "or" connector. Lists should be made consistent. A few examples of this change are shown in sections 8.3, 8.5, and 8.8.

Modify a portion of section 8.3 (End-to-end Data Protection) as shown below:

8.3 End-to-end Data Protection (Optional)

…

8.3.1 The PRACT Bit

…

8.3.1.1 Protection Information and Write Commands

…

If the namespace is formatted with protection information and the PRACT bit is set to '1', then:

1. If the namespace is formatted with Metadata Size equal to 8 (refer to Figure 115), then the logical block data is transferred from the host buffer to the controller. As the logical block data passes through the controller, the controller generates and appends protection information to the end of the logical block data, and the logical block data and protection information are written to NVM (i.e., the metadata is not resident within the host buffer)~~.~~; and
2. If the namespace is formatted with Metadata Size greater than 8, then the logical block data and the metadata are transferred from the host buffer to the controller. As the metadata passes through the controller, the controller overwrites the protection information portion of the metadata. The logical block data and metadata are written to the NVM (i.e., the metadata field remains the same size in the NVM and the host buffer). The location of the protection information within the metadata is configured when the namespace is formatted (refer to the DPS field in Figure 115).

…

8.3.1.2 The PRACT Bit and Read Commands

…

If the namespace is formatted with protection information and the PRACT bit is set to '1', then:

a) if the namespace is formatted with Metadata Size equal to 8 (refer to Figure 115), the logical block data and metadata (which in this case is, by definition, the protection information)~~,~~ is read from the NVM by the controller. As the logical block and metadata pass through the controller, the protection information is checked. If a protection information check error is detected, the command completes with the status code of the error detected (i.e., End-to-end Guard Check, End-to-end Application Tag Check or End-to-end Reference Tag Check). After processing the protection information, the controller strips it and returns the logical block data to the host (i.e., the metadata is not resident within the host buffer); and

b) if the namespace is formatted with Metadata Size greater than 8, the logical block data and the metadata, which in this case contains the protection information and additional host formatted metadata, is read from the NVM by the controller. As the logical block and metadata pass through the controller, the protection information embedded within the metadata is checked. If a protection information check error is detected, the command completes with the status code of the error detected (i.e., End-to-end Guard Check, End-to-end Application Tag Check or End-to-end Reference Tag Check). After processing the protection information, the controller passes the logical block data and metadata, with the embedded protection information unchanged, to the host (i.e., the metadata field remains the same size in the NVM as within the host buffer).

Modify a porting of section 8.5 (Virtualization Enhancements) as shown below:

8.5 Virtualization Enhancements (Optional)

…

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

- One or more primary controllers, each of which supports:
  - One or more secondary controllers;

    - A pool of unassigned Flexible Resources that supports allocation to a primary controller and dynamic assignment to its associated secondary controllers;
    - Indicate support for the Virtualization Management command in the Optional Admin Command Support (OACS) field in the Identify Controller data structure;
    - The Virtualization Management command;
    - The Primary Controller Capabilities Structure defined in Figure 110 (Identify command with CNS value of 14h);
    - The Secondary Controller List defined in Figure 111 (Identify command with CNS value of 15h); and
    - The Namespace Management capability (refer to section 8.12) ~~and Namespace Attachment commands~~;
- One or more secondary controllers; and
- Flexible Resources, each of which supports all of the following:
    - Assignment and removal by exactly one primary controller; and
    - Assignment to no more than one controller at a time.

Within an NVM subsystem that supports **…**

…

Modify a portion of section 8.8 (Reservations) (to correct list punctuation) as shown below. Other lists should be examined for equivalent changes:

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, Reservation Register command, Reservation Acquire command, and Reservation Release command;
- 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 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).

…

Modify a portion of section 8.12 (Namespace Management) as shown below:

8.12 Namespace Management (Optional)

The Namespace Management capability consist of the Namespace Management command and the Namespace Attachment command. The Namespace Management command is used to create a namespace or delete a namespace. The Namespace Attachment command is used to attach and detach controllers from a namespace. The Namespace M~~m~~anagement capability is intended for use during manufacturing or by a system administrator.

If the Namespace Management capability is supported, then the controller:

a) shall support the Namespace Management command and the Namespace Attachment command;
b) shall set bit 3 to '1' in the OACS field (refer to Figure 109); and
c) should support the Namespace Attribute Changed asynchronous event (refer to Figure 49 and section 5.21.1.11).

If a namespace is detached from a controller, then the NSID that referred to that namespace **…**