



#### **LEGAL NOTICE:**

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

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

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

**NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.:** If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: "© 2007 to 2021 NVM Express, Inc. ALL RIGHTS RESERVED." When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

#### **LEGAL DISCLAIMER:**

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

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

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

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

## NVM Express Technical Proposal for New Feature

<b>Technical Proposal ID</b>	<b>6016a</b>
<b>Change Date</b>	<b>2021-01-14</b>
<b>Builds on Specification</b>	<b>NVM Express 1.4</b> <b>NVMe-MI 1.1</b> <b>NVMe 1.4 ECN 003</b> <b>TP 6009 Additional Controller Metadata Types</b>

### Technical Proposal Author(s)

<b>Name</b>	<b>Company</b>
Nathan Obr	Microsoft

This technical proposal ports the Set/Get Features from the NVMe-MI 1.1 spec to the NVMe 1.4 spec and extends Namespace Metadata.

The latest version of NVMe-MI - TP 6009 Additional Controller Metadata Types 2019.12.09 – Preratification is incorporated.

### Revision History

<b>Revision Date</b>	<b>Change Description</b>
2020-01-06	Initial version
2020-01-13	Moved TBDA into Persistent Event Log section Included Identify Controller fields. Combined section TBDB into Enhanced Controller Metadata section and had other sections reference it. Separated Figure 271 columns into section 7.1.1 and 7.1.2.
2020-01-27	Comment resolutions: Deleted redundant Persistent Log paragraph in 5.14.1.13.1.11 (6 <sup>th</sup> paragraph) Added NVMe-MI references and clarifications to Identify Controller Data structure by 253.
2020-02-03	Changed the name of the TPAR from 'Namespace Metadata' to "Port NVMe-MI section 8 to NVMe and extend Namespace Metadata" Added NVMESD and NVMEI to NVMe-MI section 1.4 Added editors note to the changes section explicitly stating that NVMe-MI section 8 is to be deleted in its entirety. Updated all NVMe-MI references to be consistent Replaced all instances of 'Requester' to 'host' Moved Host Metadata Operation to section 8.TBD7, updated
2020-02-10	Finalized NVMe-MI section 1.4 Finalized IDENTIFY Finalized section 5.21.1.25 Removed 'Features are Independent' statement from section 8.TBD7

2020-02-24	Removed section deleted by 6009. Removed any mention of Mandatory Host Metadata Features from Figure 422 and 430. See the NVMe-MI spec instead. Added definition that the first Metadata Element descriptor contains 0h if there are no entries.
2020-03-09	Made Namespace Metadata prohibited for Admin Controllers Added new Namespace Metadata Element Types with an example for string orientation.
2020-03-16	Removed the endian-ness of the UTF-8 strings. All strings are big endian according to the NVMe base spec. Replaced meaningful new field names with generic qualifier names. Added requirement that Host Metadata is not savable.
2020-03-23	Add to section 6 with the SET FEATURES mandatory/optional table. Removed references to default values as those are already in the NVMe base spec, although almost nobody knew where to find it. Removed all <del>deleted</del> text. All sections are now in 'final' form.
2020-03-30	Removed unmodified section, irrelevant table notes Fixed wording of references clarified text. Defined the completion Dword0 bits 1 and 2. Corrected an error response from Invalid Parameter Error Response to Invalid Field In Command.
2020-04-06	Formatting and capitalization. Cosmetic changes.
2020-04-13	Added vendor specific distinction for all Host Metadata strings.
2020-04-20	Added support requirements for in-band and out-of-band. Removed superseded verbiage. Removed references to 'this specification' Made small editorial changes.
2020-06-01	Resolved conflict between power-on or reset causes the Number of Metadata Element Descriptors revert to default values or be set to 0.
2020-06-08	Added explicit feature ID name to the description of the default number of descriptors in Enhanced and Controller metadata. Removed the description of the default number of descriptors in Namespace metadata.
2020-06-15	Moved Controller Metadata Element Types from Controller Metadata section to Enhanced Controller Metadata section. Moved the overview section to be a parent of the Host Metadata features. Duplicated the Controller Metadata's "Add Entry Multiple" restriction from the overview section to the Controller Metadata section. Differentiated between default, current, and received Host Metadata Data Structures in the new Generate Default Host Metadata bit and the Element Action Field. Added Generate Default Host Metadata support bit to Identify.
2020-06-29	Removed Generate Default Host Metadata support bit to Identify. Removed Differentiation between default, current, and received Host Metadata Data Structures in the new Generate Default Host Metadata bit and the Element Action Field. Extended Generate Default Host Metadata to include 'do nothing' as an option for backward compatibility.
2020-09-08	Integrated into the NVMe Management Interface Specification, Revision 1.1 and the NVM Express Base Specification.
2020-09-10	Fixed the references to the NS Specific bit (refer to NVMe 1.4 ECN 003) in the completion queue entry. Updated each bit to use the official name.
2020-09-14	Moved "vendor specific" to be with the word "string" and not part of the number as pointed out in the Management Interface Task Group review.
2020-09-27	Integrated into the NVMe Management Interface Specification, Revision 1.1 and the NVM Express Base Specification.
2020-09-28	Removed comments and accepted all changes.
2020-11-02	Removed the Host Metadata feature definition added as part of TP 6009. Added section 5.12 to fix the field name and update the reference to the Identify Controller data structure of the NVM Express Base Specification.

2020-11-09	Fixed name of VPD Write Cycles Remaining field (added 's'). Editorial changes.
2020-11-12	Accepted all changes and removed all comments for member review.
2021-01-03	Integrated into the NVMe Base Specification and the NVMe-MI Specification, Revision 1.1.
2021-01-14	Accepted all changes and removed all comments for ratification.

## Description for NVMe 1.4 and NVMe-MI 1.1 Changes Document

This technical proposal ports the Set/Get Features from the NVMe-MI 1.1 spec to the NVMe 1.4 spec and extends Namespace Metadata.

### Markup Conventions:

Black:	Unchanged from original text in NVMe Base spec (however, hot links are removed)
Red:	Changes made to port NVMe-MI section 8 to the NVMe Base spec
Blue:	New text for extensions to Namespace Metadata and new text replacing section 8
Green:	Note to editor.

## Description of Specification Changes

### ***Make the following additions to section 1 of the NVMe-MI spec:***

#### **1.4 NVM Subsystem Architectural Model**

This specification defines an interface that may be used to manage NVM Subsystems contained within an NVMe Storage Device or NVMe Enclosure.

The NVMe Storage Device (NVMESD) bit in the NVM Subsystem Report (NVMSR) field of the Identify Controller data structure shall be set to '1' for an NVMe Storage Device. The NVMe Enclosure (NVMEE) bit in the NVM Subsystem Report (NVMSR) field of the Identify Controller data structure shall be set to '1' for an NVMe Enclosure. The NVMESD bit, the NVMEE bit, or both the NVMESD bit and the NVMEE bit shall be set to '1' (refer to the NVM Express Base Specification).

### ***Make the following additions to section 1.8.TBD1 from TP 6009 as follows:***

#### **1.8.TBD1 Host Metadata Features**

~~The Enhanced Controller Metadata feature (refer to section 8.2.3), the Controller Metadata feature (refer to section 8.2.4.), and the Namespace Metadata feature (refer to section 8.2.5).~~

### ***Make the following changes to section 5.12 as follows:***

#### **5.12 VPD Write**

The VPD Write command is used to update the Vital Product Data described in section 9.2.

After the VPD Write command has been processed without error, reading the contents of the FRU Information Device directly or a VPD Read command processed without error shall return the new VPD contents (i.e., those supplied with the VPD Write command). The data to be written to the VPD is specified in the Request Data field. VPD Write uses NVMe Management Dwords 0 and 1 as shown in Figure 107 and Figure 108.

The VPD contents should be capable of being updated at least 8 times using the VPD Write command<sup>1</sup>. If the initial value of the VPD Write Cycles Remaining field is less than 100, then the VPD Write Cycle Remaining Valid bit should be set to '1' (Refer to the VPD **Write Cycle** Information field in the **Identify Controller data structure of the NVM Express Base Specification-Figure-436**). If there is an error preventing update of the VPD contents, then the Responder responds with a Generic Error Response and VPD Writes Exceeded status.

**Make the following additions to section 6 of the NVMe-MI spec:**

#### 6.TBDA Set Features and Get Features

Figure TBDB defines features that are mandatory or optional for an I/O controller. Refer to the NVM Express Base Specification for the definition of Set Features and Get Features commands and I/O controllers.

All Feature Identifiers supported shall be supported if received in-band on an NVMe controller or received out-of-band on a Management Endpoint unless otherwise stated.

**Figure TBDB: I/O Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>	Logged in Persistent Event Log <sup>1</sup>
Enhanced Controller Metadata	M	O
Controller Metadata	M	O
Namespace Metadata	M	O
Notes: 1. O = Optional, M = Mandatory		

Figure TBDC defines features that are mandatory or optional for an administrative controller. Refer to the NVM Express Base specification for the definition of Set Features and Get Features commands and administrative controller.

**Figure TBDC: Administrative Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>	Logged in Persistent Event Log <sup>1</sup>
Enhanced Controller Metadata	M	O
Controller Metadata	M	O
Namespace Metadata	O	O
Notes: 1. O = Optional, M = Mandatory		

**Make the following removals from the NVMe-MI spec:**

Remove all of the current text in NVMe-MI section 8. The contents of section 8 are to be incorporated into the NVMe Base spec in the following sections below.

**Make the following addition to section 5 of the NVMe Base spec:**

The following is from NVMe 1.4 ECN 003 for reference to the changes in this TP for the command completion of a Get Features command:

#### 5.13.2 Command Completion

<sup>1</sup> NVM Express Management Interface Specification, Revision 1.0a and prior recommended that VPD contents should be capable of being updated at least 100 times using the VPD Write command.

Upon completion of the Get Features command, the controller posts a completion queue entry to the Admin Completion Queue. If the Select field is not set to 11b, then Dword 0 of the completion queue entry may contain feature-dependent information (refer to section 5.21.1).

If the Select field is set to 11b, then Figure NEW describes the contents of Dword 0 of the completion queue entry.

**Figure NEW - Completion Queue Entry Dword 0 when Select=11b**

Bits	Description
31:3	Reserved
2	<b>Changeable:</b> If set to '1', then the feature values are changeable. If cleared to '0', then the feature values are not changeable.
1	<b>NS Specific:</b> If set to '1', then the Feature Identifier is namespace specific and settings are applied to individual namespaces. If cleared to '0', then the Feature Identifier is not namespace specific and its settings apply to the entire controller.
0	<b>Saveable:</b> If set to '1', then the feature values are saveable. If cleared to '0', then the feature values are not saveable.

### 5.15.2.2 Identify Controller data structure (CNS 01h)

The Identify Controller data structure (refer to Figure 247) is returned to the host for this controller.

**Figure 247: Identify – Identify Controller Data Structure**

Bytes	O/M <sup>1</sup>	Description								
252:240		Reserved for the NVMe Management Interface.								
253	M	<b>NVM Subsystem Report (NVMSR):</b> This field reports information associated with the NVM subsystem. If the controller is compliant to the Management Interface Specification, then at least one bit in this field is set to '1'. If the NVM subsystem does not support the Management Interface Specification then this field shall be cleared to 0h. Refer to the NVMe Management Interface Specification.								
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1</td><td><b>NVMe Enclosure (NVMEE):</b> If set to '1', then the NVM subsystem is part of an NVMe Enclosure. If cleared to '0', then the NVM Subsystem is not part of an NVMe Enclosure.</td></tr><tr><td>0</td><td><b>NVMe Storage Device (NVMESD):</b> If set to '1', then the NVM subsystem is part of an NVMe Storage Device. If cleared to '0', then the NVM subsystem is not part of an NVMe Storage Device.</td></tr></table>	Bits	Description	7:2	Reserved	1	<b>NVMe Enclosure (NVMEE):</b> If set to '1', then the NVM subsystem is part of an NVMe Enclosure. If cleared to '0', then the NVM Subsystem is not part of an NVMe Enclosure.	0	<b>NVMe Storage Device (NVMESD):</b> If set to '1', then the NVM subsystem is part of an NVMe Storage Device. If cleared to '0', then the NVM subsystem is not part of an NVMe Storage Device.
		Bits	Description							
		7:2	Reserved							
1	<b>NVMe Enclosure (NVMEE):</b> If set to '1', then the NVM subsystem is part of an NVMe Enclosure. If cleared to '0', then the NVM Subsystem is not part of an NVMe Enclosure.									
0	<b>NVMe Storage Device (NVMESD):</b> If set to '1', then the NVM subsystem is part of an NVMe Storage Device. If cleared to '0', then the NVM subsystem is not part of an NVMe Storage Device.									

**Figure 247: Identify – Identify Controller Data Structure**

Bytes	O/M <sup>1</sup>	Description								
254	M	<b>VPD Write Cycle Information (VWCI):</b> This field indicates information about remaining number of times that VPD contents are able to be updated using the VPD Write command. Refer to the NVMe Management Interface Specification for details on VPD contents and the VPD Write command.								
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7</td><td><b>VPD Write Cycle Remaining Valid (VWCRV):</b> If this bit is set to '1', then the VPD Write Cycles Remaining field is valid. If this bit is cleared to '0', then the VPD Write Cycles Remaining field is invalid and cleared to '0'.</td></tr><tr><td>6:0</td><td><b>VPD Write Cycles Remaining (VWCR):</b> If the VPD Write Cycle Remaining Valid bit is set to '1', then this field contains a value indicating the remaining number of times that VPD contents are able to be updated using the VPD Write command. If this field is set to 7Fh, then the remaining number of times that VPD contents are able to be updated using the VPD Write command is greater than or equal to 7Fh.  If the VPD Write Cycle Remaining Valid bit is cleared to '0', then this field is not valid and shall be cleared to a value of 0h.</td></tr></table>	Bits	Description	7	<b>VPD Write Cycle Remaining Valid (VWCRV):</b> If this bit is set to '1', then the VPD Write Cycles Remaining field is valid. If this bit is cleared to '0', then the VPD Write Cycles Remaining field is invalid and cleared to '0'.	6:0	<b>VPD Write Cycles Remaining (VWCR):</b> If the VPD Write Cycle Remaining Valid bit is set to '1', then this field contains a value indicating the remaining number of times that VPD contents are able to be updated using the VPD Write command. If this field is set to 7Fh, then the remaining number of times that VPD contents are able to be updated using the VPD Write command is greater than or equal to 7Fh.  If the VPD Write Cycle Remaining Valid bit is cleared to '0', then this field is not valid and shall be cleared to a value of 0h.		
		Bits	Description							
		7	<b>VPD Write Cycle Remaining Valid (VWCRV):</b> If this bit is set to '1', then the VPD Write Cycles Remaining field is valid. If this bit is cleared to '0', then the VPD Write Cycles Remaining field is invalid and cleared to '0'.							
6:0	<b>VPD Write Cycles Remaining (VWCR):</b> If the VPD Write Cycle Remaining Valid bit is set to '1', then this field contains a value indicating the remaining number of times that VPD contents are able to be updated using the VPD Write command. If this field is set to 7Fh, then the remaining number of times that VPD contents are able to be updated using the VPD Write command is greater than or equal to 7Fh.  If the VPD Write Cycle Remaining Valid bit is cleared to '0', then this field is not valid and shall be cleared to a value of 0h.									
255	M	<b>Management Endpoint Capabilities (MEC):</b> This field indicates the capabilities of the Management Endpoint in the NVM subsystem. Refer to the NVMe Management Interface Specification for details.								
		<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:2</td><td>Reserved</td></tr><tr><td>1</td><td><b>PCIe Port Management Endpoint (PCIEME):</b> If set to '1', then the NVM subsystem contains a Management Endpoint on a PCIe port.</td></tr><tr><td>0</td><td><b>SMBus/I2C Port Management Endpoint (SMBUSME):</b> If set to '1', then the NVM subsystem contains a Management Endpoint on an SMBus/I2C port.</td></tr></table>	Bits	Description	7:2	Reserved	1	<b>PCIe Port Management Endpoint (PCIEME):</b> If set to '1', then the NVM subsystem contains a Management Endpoint on a PCIe port.	0	<b>SMBus/I2C Port Management Endpoint (SMBUSME):</b> If set to '1', then the NVM subsystem contains a Management Endpoint on an SMBus/I2C port.
		Bits	Description							
		7:2	Reserved							
1	<b>PCIe Port Management Endpoint (PCIEME):</b> If set to '1', then the NVM subsystem contains a Management Endpoint on a PCIe port.									
0	<b>SMBus/I2C Port Management Endpoint (SMBUSME):</b> If set to '1', then the NVM subsystem contains a Management Endpoint on an SMBus/I2C port.									

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

### 5.21.1 Feature Specific Information

**Figure 271: Set Features – Feature Identifiers**

Feature Identifier	Current Setting Persists Across Power Cycle and Reset <sup>2</sup>	Uses Memory Buffer for Attributes	Feature Name
00h	Reserved		
01h	No	No	Arbitration
03h	Yes	Yes	LBA Range Type
04h	No	No	Temperature Threshold
05h	No	No	Error Recovery
06h	No	No	Volatile Write Cache
07h	No	No	Number of Queues

**Figure 271: Set Features – Feature Identifiers**

Feature Identifier	Current Setting Persists Across Power Cycle and Reset <sup>2</sup>	Uses Memory Buffer for Attributes	Feature Name
08h	No	No	Interrupt Coalescing
09h	No	No	Interrupt Vector Configuration
0Ah	No	No	Write Atomicity Normal
0Bh	No	No	Asynchronous Event Configuration
0Ch	No	Yes	Autonomous Power State Transition
0Dh	No <sup>3</sup>	No <sup>4</sup>	Host Memory Buffer
0Eh	No	Yes	Timestamp
0Fh	No	No	Keep Alive Timer
10h	Yes	No	Host Controlled Thermal Management
11h	No	No	Non-Operational Power State Config
12h	Yes	No	Read Recovery Level Config
13h	No	Yes	Predictable Latency Mode Config
14h	No	No	Predictable Latency Mode Window
15h	No	No	LBA Status Information Report Interval
16h	No	Yes	Host Behavior Support
17h	Yes	No	Sanitize Config
18h	No	No	Endurance Group Event Configuration
19h to 77h	Reserved		
78h to 7Ch	Refer to the NVMe Management Interface Specification for definition. Reserved for Management Features.		
7Dh	No	Yes	Enhanced Controller Metadata
7Eh	No	Yes	Controller Metadata
7Fh	No	Yes	Namespace Metadata
80h to BFh			Command Set Specific (Reserved)
C0h to FFh			Vendor Specific <sup>1, 5</sup>
<b>NOTES:</b> <ol style="list-style-type: none"> <li>The behavior of a controller in response to an inactive namespace ID to a vendor specific Feature Identifier is vendor specific.</li> <li>This column is only valid if the feature is not saveable (refer to section 7.8). If the feature is saveable, then this column is not used.</li> <li>The controller does not save settings for the Host Memory Buffer feature across power states and reset events, however, host software may restore the previous values. Refer to section 8.9.</li> <li>The feature does not use a memory buffer for Set Features commands and does use a memory buffer for Get Features commands. Refer to section 8.9.</li> <li>Selection of a UUID may be supported. Refer to section 8.24.</li> </ol>			



### 5.21.1. TBD7 Host Metadata (Feature Identifier 7Dh), (Feature Identifier 7Eh), (Feature Identifier 7Fh)

The Host Metadata features are the Enhanced Controller Metadata feature (Feature Identifier 7Dh), the Controller Metadata feature (Feature Identifier 7Eh), and the Namespace Metadata feature (Feature Identifier 7Fh).

If a Get Features command specifying one of the Host Metadata features with the SEL field set to 011b (i.e., Supported Capabilities) for these Feature Identifiers is submitted, then the Saveable bit in Dword 0 of the corresponding completion queue entry shall be cleared to '0' (i.e., refer to section 7.8), and the Changeable bit in Dword 0 of the corresponding completion entry shall be set to '1'.

If a Get Features command specifying one of the Host Metadata features, the controller shall perform additional actions specified in Figure TBD8.

**Figure TBD8: Get Features – Command Dword 11**

Bits	Description
31:01	Reserved
00	<p><b>Generate Default Host Metadata (GDHM):</b> If set to '1' then the controller shall generate a number of vendor specific strings for the Element Types of the specified Host Metadata feature value.</p> <p>If the generated vendor specific string's Metadata Element Descriptor does not exist for the Host Metadata Data Structure that contains the default value of the specified Host Metadata Feature value, then the controller shall create the Metadata Element Descriptor in the Host Metadata Data Structure that contains the default value with the generated vendor specific string.</p> <p>If the generated vendor specific string's Metadata Element Descriptor does exist for the Host Metadata Data Structure that contains the default value of the specified Host Metadata Feature value, then the controller shall replace the Metadata Element Descriptor with the generated vendor specific string.</p> <p>If the number of vendor specific strings generated is 0h then the default value for the Number of Metadata Element Descriptors for the specified Host Metadata feature shall be 0h. If the number of vendor specific strings generated is not 0h, then the Host Metadata Data Structure that contains the default value for the Number of Metadata Element Descriptors of the specified Host Metadata Feature value shall be the number of vendor specific strings created.</p> <p>If cleared to '0' then the device shall not generate any vendor specific strings for the Element Types of the specified Host Metadata feature.</p>

The host issues a Set Features command specifying one of the Host Metadata features containing a Host Metadata data structure (refer to Figure TBD3). The host receives a Host Metadata data structure via the Get Features command. The content of the strings in the Host Metadata data structure are vendor specific.

The Action is specified in Command Dword 11 as shown in Figure TBD2.

**Figure TBD2: Set Features – Command Dword 11**

Bits	Description
31:15	Reserved

**Figure TBD2: Set Features – Command Dword 11**

Bits	Description										
14:13	<p><b>Element Action (EA):</b> This field specifies the action to perform on the specified Host Metadata Feature value for each Metadata Element Descriptor data structure contained in the Host Metadata data structure. This field shall be cleared to 0h for a Get Features command.</p> <table border="1"> <thead> <tr> <th>Value</th><th>Definition</th></tr> </thead> <tbody> <tr> <td>00b</td><td>Add/Replace Entry</td></tr> <tr> <td>01b</td><td>Delete Entry Multiple</td></tr> <tr> <td>10b</td><td>Add Entry Multiple</td></tr> <tr> <td>11b</td><td>Reserved</td></tr> </tbody> </table> <p>If the Element Action field is cleared to 00b (Add/Replace Entry) and the Metadata Element Descriptor with the specified Element Type (refer to Figure TBD4) does not exist in the specified Host Metadata Feature value, then the Controller shall create the descriptor in the specified Host Metadata Feature value with the value in the Host Metadata data structure.</p> <p>If the Element Action field is set to 00b (Add/Replace Entry) and one Metadata Element Descriptor with the specified Element Type exists in the specified Host Metadata Feature value, then the Controller shall replace with the value in the specified Host Metadata data structure.</p> <p>If the Element Action field is cleared to 00b (Add/Replace Entry) and the Feature Identifier field is set to Enhanced Controller Metadata, then the controller shall abort the Set Features command with Invalid Field in Command status and shall not change any Host Metadata Feature value.</p> <p>If the Element Action field is set to 01b (Delete Entry Multiple), then the Controller shall delete all the specified Metadata Element Descriptors from the specified Host Metadata Feature value, if any. If none of the specified Metadata Element Descriptors are present in the specified Host Metadata Feature value, then the controller shall complete the Set Features command with a status of Successful Completion and shall not change any Host Metadata Feature value.</p> <p>If the Element Action field is set to 10b (Add Entry Multiple), the Feature Identifier field is set to Enhanced Controller Metadata, and no Metadata Element Descriptor with the specified Element Type exists in the Enhanced Controller Metadata Feature value, then the controller shall create new Metadata Element Descriptors in the Enhanced Controller Metadata Feature value with the Element Type and the value specified in the Host Metadata data structure.</p> <p>If the Element Action field is set to 10b (Add Entry Multiple), the Feature Identifier field is set to Enhanced Controller Metadata, and one or more Metadata Element Descriptors with the specified Element Type exists in the Enhanced Controller Metadata Feature value, then the controller shall add the specified Metadata Element to the Enhanced Controller Metadata Feature value and shall not modify any existing Metadata Element Descriptors.</p> <p>If the Element Action field is set to 10b (Add Entry Multiple) and the Feature Identifier field is not set to Enhanced Controller Metadata, then the controller shall abort the Set Features command with status Invalid Field in Command and shall not change the Host Metadata Feature value.</p>	Value	Definition	00b	Add/Replace Entry	01b	Delete Entry Multiple	10b	Add Entry Multiple	11b	Reserved
Value	Definition										
00b	Add/Replace Entry										
01b	Delete Entry Multiple										
10b	Add Entry Multiple										
11b	Reserved										
12:00	Reserved										

Metadata Element Descriptors may be added, replaced, or deleted based on the action specified in the Element Action field. Modification of the Host Metadata Feature value shall be performed by the controller in an atomic manner.

If a Set Features command is submitted for a Host Metadata Feature, a Host Metadata data structure, defined in Figure TBD3, is transferred in the data buffer for the command. The Host Metadata data structure is 4 KiB in size and contains zero or more Metadata Element Descriptors. If host software attempts to add

or replace a Metadata Element that causes the Host Metadata Feature value of the specified feature to grow larger than 4 KiB, then the controller shall abort the command with an Invalid Field In Command.

If the host receives a Host Metadata data structure via the Get Features command, then all of the Metadata Element Descriptors present for the specified feature are added to a Host Metadata data structure (refer to Figure TBD3) and returned in the data buffer for that command. The data buffer size is equal to the size of the Host Metadata data structure that is 4 KiB in size.

**Figure TBD3: Host Metadata Data Structure**

Bytes	Description
00	<b>Number of Metadata Element Descriptors:</b> This field contains the number of Metadata Element descriptors in the data structure.
01	Reserved
x:02	<b>Metadata Element Descriptor 0:</b> This field contains the first Metadata Element descriptor or 0h if there are no entries.
y:x+1	<b>Metadata Element Descriptor 1:</b> This field contains the second Metadata Element descriptor or 0h if there is only 1 entry.
...	...
4095:z	<b>Metadata Element Descriptor N:</b> This field contains the (N+1)th Metadata Element descriptor or 0h if there are fewer than N+1 entries.

If the Feature Identifier field specifies Controller Metadata or Namespace Metadata, then the Host Metadata data structure may contain at most one Metadata Element Descriptor of each Element Type. If the Feature Identifier field specifies Enhanced Controller Metadata, then a Host Metadata data structure may contain more than one Metadata Element Descriptor of each Element Type. Each Metadata Element Descriptor contains the data structure shown in Figure TBD4.

**Figure TBD4: Metadata Element Descriptor**

Bit	Description
31 + (Element Length*8) :32	<b>Element Value (EVAL):</b> This field specifies the value for the element.
31:16	<b>Element Length (ELEN):</b> This field specifies the length of the Element Value field in bytes. This field shall be 0h when deleting an entry (i.e., EA = 01b). This field should be non-zero when adding/updating and entry (i.e., EA = 00b).
15:12	Reserved
11:08	<b>Element Revision (ER):</b> This field specifies the revision of this element value. Unless specified otherwise elsewhere in this specification, all Metadata Element Descriptors shall clear this field to 0h.
07:06	Reserved

**Figure TBD4: Metadata Element Descriptor**

Bit	Description								
05:00	<b>Element Type (ET):</b> This field specifies the type of metadata stored in the descriptor.								
	<table><tr><th>Value</th><th>Definition</th></tr><tr><td>00h</td><td>Reserved</td></tr><tr><td>01h to 017h</td><td>Element Types defined by this specification. Enhanced Controller Metadata Element and Controller Metadata Element types are defined in Figure TBD5. Namespace Metadata Element types are defined in Figure TBD6.</td></tr><tr><td>18h to 1Fh</td><td>Vendor Specific</td></tr></table>	Value	Definition	00h	Reserved	01h to 017h	Element Types defined by this specification. Enhanced Controller Metadata Element and Controller Metadata Element types are defined in Figure TBD5. Namespace Metadata Element types are defined in Figure TBD6.	18h to 1Fh	Vendor Specific
Value	Definition								
00h	Reserved								
01h to 017h	Element Types defined by this specification. Enhanced Controller Metadata Element and Controller Metadata Element types are defined in Figure TBD5. Namespace Metadata Element types are defined in Figure TBD6.								
18h to 1Fh	Vendor Specific								

**5.21.1.TBD7.1 Enhanced Controller Metadata (Feature Identifier 7Dh)**

This feature is used to store metadata about the host platform in an NVM subsystem for later retrieval.

The metadata element types defined in Figure TBD5 are used by this feature.

**Figure TBD5: Controller Metadata Element Types**

Value	Definition
00h	Reserved
01h	<b>Operating System Controller Name:</b> The name of the Controller in the operating system as a UTF-8 string.
02h	<b>Operating System Driver Name:</b> The name of the driver in the operating system as a UTF-8 string.
03h	<b>Operating System Driver Version:</b> The version of the driver in the operating system as a UTF-8 string.
04h	<b>Pre-boot Controller Name:</b> The name of the Controller in the pre-boot environment as a UTF-8 string.
05h	<b>Pre-boot Driver Name:</b> The name of the driver in the pre-boot environment as a UTF-8 string.
06h	<b>Pre-boot Driver Version:</b> The version of the driver in the pre-boot environment as a UTF-8 string.
07h	<b>System Processor Model:</b> The model of the processor as a UTF-8 string.
08h	<b>Chipset Driver Name:</b> The chipset driver name as a UTF-8 string.
09h	<b>Chipset Driver Version:</b> The chipset driver version as a UTF-8 string.
0Ah	<b>Operating System Name and Build:</b> The operating system name and build as a UTF-8 string.

**Figure TBD5: Controller Metadata Element Types**

Value	Definition
0Bh	<b>System Product Name:</b> The system product name as a UTF-8 string.
0Ch	<b>Firmware Version:</b> The host firmware (e.g., UEFI) version as a UTF-8 string.
0Dh	<b>Operating System Driver Filename:</b> The operating system driver filename as a UTF-8 string.
0Eh	<b>Display Driver Name:</b> The display driver name as a UTF-8 string.
0Fh	<b>Display Driver Version:</b> The display driver version as a UTF-8 string.
10h	<b>Host-Determined Failure Record:</b> A failure record (e.g., the reason the host has flagged a failure for an NVMe Storage Device (refer to the NVMe Management Interface Specification) FRU which may be used for failure analysis) as a UTF-8 string.
11h to 17h	Reserved
18h to 1Fh	Vendor Specific

Refer to section 5.21.1.TBD7 for the definitions of Command Dword 11 and the Host Metadata Data Structure.

The default value for the Number of Metadata Element Descriptors of the Enhanced Controller Metadata Feature shall be 0h on a Controller Level Reset.

If a Get Features command with the SEL field set to 011b (i.e., Supported Capabilities) with the Enhanced Controller Metadata Feature value is submitted, then the NS Specific bit in Dword 0 of the corresponding completion queue entry shall be cleared to '0'.

#### **5.21.1.TBD7.2 Controller Metadata (Feature Identifier 7Eh)**

This feature is used to store metadata about the host platform in an NVM subsystem for later retrieval.

The Controller Metadata Feature provides backward compatibility with Management Controllers (refer to the NVMe Management Interface Specification) compliant to version 1.1 and earlier versions of the NVMe Management Interface Specification.

If a controller supports both the Enhanced Controller Metadata Feature and the Controller Metadata Feature, then the Controller Metadata Feature should not be used by the host.

The metadata element types defined in Figure TBD5 are used by this feature.

Refer to section 5.21.1.TBD7 for the definitions of Command Dword 11 and the Host Metadata Data Structure.

If a Set Features command's Element Action field of Command Dword 11 is set to 10b (Add Entry Multiple), then the controller shall abort the command with status Invalid Field in Command and shall not change the Host Metadata Feature value.

The default value for the Number of Metadata Element Descriptors of the Controller Metadata Feature shall be 0h on a Controller Level Reset.

If a Get Features command with the SEL field set to 011b (i.e., Supported Capabilities) with the Controller Metadata Feature value is submitted, then the NS Specific bit in Dword 0 of the corresponding completion queue entry shall be cleared to '0'.

### 5.21.1.TBD7.3 Namespace Metadata (Feature Identifier 7Fh)

This feature is used to store metadata about a namespace associated with a controller in the NVM subsystem for later retrieval. This feature is namespace specific. The Add Entry Multiple action is prohibited for this feature.

**Figure TBD6: Namespace Metadata Element Types**

Value	Definition
00h	Reserved
01h	<b>Operating System Namespace Name:</b> The name of the namespace in the operating system as a UTF-8 string.
02h	<b>Pre-boot Namespace Name:</b> The name of the namespace in the pre-boot environment as a UTF-8 string.
03h	<b>Operating System Namespace Name Qualifier 1:</b> The first qualifier of the Operating System Namespace Name as a UTF-8 string.
04h	<b>Operating System Namespace Name Qualifier 2:</b> The second qualifier of the Operating System Namespace Name as a UTF-8 string.
05h to 17h	Reserved
18h to 1Fh	Vendor Specific

Refer to section 5.21.1.TBD7 for the definitions of Command Dword 11 and the Host Metadata Data Structure.

If a Get Features command with the SEL field set to 011b (i.e., Supported Capabilities) with the Namespace Metadata Feature value is submitted, then the NS Specific bit in Dword 0 of the corresponding completion queue entry shall be set to '1'.

**Make the following addition to section 7 of the NVMe Base spec:**

#### 7.1.1 I/O Controller

Figure 422 and Figure 423 define features that are mandatory, optional, and prohibited for an I/O controller.

**Figure 422: I/O Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>	Logged in Persistent Event Log <sup>1</sup>
Arbitration	M	O
Power Management	M	NR
LBA Range Type	O	NR
Temperature Threshold	M	O
Error Recovery	M	O
Volatile Write Cache	O	O
Number of Queues	M	O
Interrupt Coalescing	NOTE 2	O
Interrupt Vector Configuration	NOTE 2	O
Write Atomicity Normal	M	O
Asynchronous Event Configuration	M	NR
Autonomous Power State Transition	O	O

**Figure 422: I/O Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>	Logged in Persistent Event Log <sup>1</sup>
Host Memory Buffer	O	O
Timestamp	O	P
Keep Alive Timer	O	O
Host Controlled Thermal Management	O	O
Non-Operational Power State Config	O	O
Read Recovery Level Config	O	O
Predictable Latency Mode Config	O	O
Predictable Latency Mode Window	O	P
LBA Status Information Attributes	O	O
Host Behavior Support	O	O
Sanitize Config	O	O
Endurance Group Event Configuration	O	O
Enhanced Controller Metadata	O <sup>3</sup>	O <sup>3</sup>
Controller Metadata	O <sup>3</sup>	O <sup>3</sup>
Namespace Metadata	O <sup>3</sup>	O <sup>3</sup>
Notes: 1. O = Optional, M = Mandatory, P = Prohibited, NR = Not Recommended 2. The feature is mandatory for NVMe over PCIe. This feature is not supported for NVMe over Fabrics. 3. This feature is optional for NVM subsystems that do not implement a Management Endpoint. For NVM subsystems that implement any Management Endpoint refer to the NVMe Management Interface Specification.		

### 7.1.1 Administrative Controller

Figure 430 and Figure 431 defines features that are mandatory, optional, and prohibited for an administrative controller. If any feature is supported, then the Set Features and Get Features commands shall be supported.

**Figure 430: Administrative Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>	Logged in Persistent Event Log <sup>1</sup>
Arbitration	P	P
Power Management	O	NR
LBA Range Type	P	P
Temperature Threshold	O	O
Error Recovery	P	P
Volatile Write Cache	P	P
Number of Queues	P	P
Interrupt Coalescing	NOTE 2	NOTE 2
Interrupt Vector Configuration	NOTE 2	NOTE 2
Write Atomicity Normal	P	P
Asynchronous Event Configuration	O <sup>3</sup>	NR
Autonomous Power State Transition	O	O
Host Memory Buffer	O	O
Timestamp	O	P
Keep Alive Timer	O	O
Host Controlled Thermal Management	O	O
Non-Operational Power State Config	O	O
Read Recovery Level Config	O	O
Predictable Latency Mode Config	O	P

**Figure 430: Administrative Controller – Feature Support**

Feature Name	Feature Support Requirements <sup>1</sup>	Logged in Persistent Event Log <sup>1</sup>
Predictable Latency Mode Window	O	O
LBA Status Information Attributes	P	O
Host Behavior Support	O	O
Sanitize Config	O	O
Endurance Group Event Configuration	O	O
Enhanced Controller Metadata	O <sup>4</sup>	O <sup>4</sup>
Controller Metadata	O <sup>4</sup>	O <sup>4</sup>
Namespace Metadata	O <sup>4</sup>	O <sup>4</sup>
Notes: 1. O = Optional, M = Mandatory, P = Prohibited, NR = Not Recommended 2. The feature is optional for NVMe over PCIe. This feature is not supported for NVMe over Fabrics. 3. Mandatory if Telemetry Log, Firmware Commit or SMART/Health Critical Warnings are supported. 4. This feature is optional for NVM subsystems that do not implement a Management Endpoint. For NVM subsystems that implement any Management Endpoint refer to the NVMe Management Interface Specification.		