LEGAL NOTICE:

This NVM Express Management Interface revision 1.1 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 Management Interface revision 1.1 technical proposal subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

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

LEGAL DISCLAIMER:

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

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

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

NVM Express Technical Proposal for New Feature

Technical Proposal ID	**6009 – Additional Controller Metadata Types**
Change Date	**2020-03-23**
Builds on Specification	**NVMe-MI 1.1**

Technical Proposal Author(s)

Name	Company
Paul Suhler	Micron Technology

Adds additional controller metadata types useful for creating an installation record and a host-determined failure record in the Set Features event in the Persistent Event log page. Adds the new Enhanced Controller Metadata feature.

Revision History

Revision Date	Change Description
2018-08-20	Initial version
2018-09-12	Field name correction from Harvey Newman. Added element type for Host-Determined Failure Record.
2018-10-24	Corrected typo in element type name
2018-12-10	Changes from 2018-12-03 WG meeting: • Changed “Bios Version” to “host Firmware Version”. • Added capabilities bit to indicate whether more than more than one Metadata Element of the same type may be specified by a Set Features command.
2018-12-14	Changes from 2018-12-10 WG meeting: • Added Element Action of Add/Update Entry Multiple
2019-01-28	Changes from 2018-12-17 WG meeting: • Updated Element Action definitions • Added reference entry for Redfish Resource and Schema Guide • Added possible names for element types from Redfish
2019-03-10	Updated figure numbers to match NVMe-MI 1.1 2019.02.19. Changes from 2019-03-04 WG meeting: • Changed Host CPU Model to Host System Processor Model.
2019-03-31	Changes from 2019-03-11 WG meeting: • Added first draft of a model section; location in spec is TBD. • Added question in comment reply for Figure 137. Please review. • Added answers to questions in comment for Figure 141. Please review. • Decided not to make changes to namespace metadata. • Decided not to mention any use of structuring of data (e.g., via XML) inside the UTF-8 string of any metadata entry.
2019-04-08	Changes from 2019-04-01 WG meeting: • Added to section 8 a statement that Management Enhancements may be supported by controllers in subsystems that do not support management endpoints. • Added a list of items to be resolved before exiting phase 2.

2019-04-09	Changes from 2019-04-08 WG meeting: • Added bits to Figure 136 to advertise support for the two features defined in MI. • Moved “model section” text into a new subsection in 8.2.1. • Fixed reserved bytes row in Figure 136. • Changed Element Action values (Figure 138) from binary to hexadecimal, to match 1.1 changes in the following text. This may also be handled during 45-day review editorial comment resolution. • Corrected all figure numbers to match NVMe 1.1.
2019-04-29	Changes from 2019-04-29 WG meeting: • Updated original text to match NVMe-MI 1.1 “review & resolved” revision. • Added notes for Phase 3 changes.
2019-05-16	Initial Phase 3 revision. Changes from 2019-05-13 WG meeting, including Austin’s e-mailed comments: • Reverted Element Action field to two bits. • Removed “Host” from name of most element types. • Added comments to record meeting discussion.
2019-06-10	More changes from 2019-05-13 WG meeting: • Changed element action “update” to “replace”, including in existing text. • Clarified the difference between “Controller Metadata” and “Controller Metadata structure”. • Modified Add Multiple element action to be Add Entry Multiple, to avoid replacing elements that may have been set by a different host.
2019-06-21	Changes from 2019-06-17 WG meeting: • Fixed spec references in first page to be MI, not the base spec. • Renamed Delete Entry to Delete Entry Multiple. • Reorganized section 8.2 to reduce redundancy and increase clarity.
2019-06-24	Changes from discussions outside the WG meetings: • Noted that a new TP for reporting support of feature IDs may affect changes proposed here for Figure 136. • Re-titled Figure 136. • Moved Same-type metadata element descriptors to bit 0 of Management Capabilities. • Moved discussion of interactions with the Persistent Event Log to 8.2.1, to be generic for both feature IDs. • Added comment to document Jim Hatfield’s question about whether there are any requirements around ordering of metadata elements when they are added.
2019-07-08	Changes from 2019-07-08 WG meeting: • Added Management Capabilities Valid bit to indicate validity of the new Management Capabilities byte. This bit shall be ‘1’ for controllers compliant with NVMe-MI 1.2 and later. • Added statement about whether each feature ID is mandatory for administrative controllers and for I/O controllers. This revision should be ready for member review.
2019-07-21	Changes from Austin Bolen’s comments and 2019-07-15 WG meeting: • Deleted long-standing statement that setting these features does not modify controller behavior. • Deleted Same-Type Metadata Element Descriptors field and keyed support to bit 7. Reworked text keying support to spec revision. • Revised reset behavior to clear the Number of Metadata Element Descriptors field. • Moved reset behavior from the two different features’ sections into the Host Metadata Overview section. • Added “datastore” to host metadata in an attempt to better distinguish it from data sent or received via the Set/Get Features commands.

2019-07-22	Changes from 2019-07-22 WG meeting: • Moved controller metadata element types allowing multiple entries per type into a new feature ID. • Revised references to host metadata datastore to use feature value. • Changed reset behavior to state that feature is not saveable.
2019-08-19	Changes from 2019-07-29 WG meeting: • Reinstated new feature ID for Enhanced Controller Metadata. • Removed changes to Identify Controller data structure. • Uses the term “host metadata feature” to refer to all three feature IDs. • Added Primary Port Controller definition. • Added Reference column to Figure 137. • Showed detailed changes from NVMe-MI 1.1.
2019-08-20	Changes from 2019-08-19 WG meeting and Mike Allison’s e-mail: • Removed “Primary Port Controller”. • Made capitalization of feature names consistent. • Replaced “not allowed” with “prohibited”. • Added statements explaining intention for Enhanced Controller Metadata Feature to supersede Controller Metadata Feature. • Made all references to NVM Express Base Specification consistent.
2019-08-26	Member review candidate. Changes from 2019-08-19 WG meeting: • Modified mandatory/optional and note 3 in Figure 137. • In 8.2.1 clarified that string is a UTF-8 string, and removed parenthetical example.
2019-09-09	Member review candidate. Changes from 2019-09-09 TG meeting: • Make Enhanced Controller Metadata feature and Controller Metadata feature mandatory.
2019-10-02	Proposed resolutions to member review comments.
2019-10-09	Changes from 2019-10-07 TG meeting, including: • Replaced “Host Metadata feature” with “Host Metadata Feature”. • Deleted sentence defining “host metadata feature value”, as it is redundant. Feature values are explained in the Base Spec. • Renamed Add Entry Multiple to Add/Replace Entry Multiple and added explanation that existing descriptors are replaced. This had been implied.
2019-10-15	Changes from further reviews: • Revert “Add/Replace Entry Multiple” action name to “Add Entry Multiple”. • Revert second paragraph for Add Entry Multiple to the description in the 2019.06.10 version.
2019-10-24	Accepted changes from first member review (last documented in the 2019-10-15 revision. Resolved new review comments from Austin Bolen.
2019-10-29	Resolved more review comments from Austin Bolen.
2019-10-31	Revision for Integration. One more resolved review comment from Austin Bolen.
2019-11-18	Resolution of TBDs
2019-12-05	Retitled for ratification. Fixed heading.
2019-12-09	Added back the requirement from NVMe-MI 1.1 to clear Number of Metadata Element Descriptors to 0h on a Controller Level Reset.
2020-01-10	Changed dates to reflect 2020.
2020-02-24	Removing the “specification” to “Specification” changes when referring to the NVMe specification. Changed “exists” to “exist” to match editorial change during integration.
2020-03-23	Ratified

Description for NVMe-MI Changes Document

Additional Controller Metadata Types

- Adds additional Controller Metadata types useful for creating an installation record and a host-determined failure record in the Set Features event in the Persistent Event log page.
- Moves text that applies to both controller and namespace metadata into a common section.
- Adds a new feature, Enhanced Controller Metadata, which allows multiple metadata elements of the same type.
- Renamed Add/Update Entry element action to Add Entry. Renamed Delete Entry element action to Delete Entry Multiple and redefined it to delete one or more entries. Earlier behavior (deleting a single entry) works as previously defined.
- Adds the Add Entry Multiple element action to set multiple metadata elements of the same type, to be used only for the Enhanced Controller Metadata feature.
- References:
  - NVMe-MI 1.1 section 8
  - Technical Proposal 6009

Background (not part of specification changes)

Failure analysis of an SSD relies upon knowing the environment in which it failed, such as the product name, system hardware, operating system, drivers, etc. If a persistent installation record were placed in the device when it was installed and when elements of the record changed, it could provide the context for the failure when read by a host performing failure analysis. In addition to the installation record, a host-determined failure record would record the reason for removal of the SSD, such as a textual description of the failure.

This proposal defines minor extensions to existing NVMe-MI capabilities to enable hosts to create installation records and host-determined failure records in SSDs.

The Set Features and Get Features commands presenting and retrieving the Controller Metadata feature may be issued by either a host over a PCIe bus (or other fabric) or a baseboard management controller (BMC) over SMBus/I2C.

Description of Specification Changes:

- Extended features to facilitate using them to save host environment information in the controller. These are usable over any interface. The information may be retrieved via the Persistent Event Log (refer to NVMe Base Specification 1.4).
- Added a new feature, Enhanced Controller Metadata (Feature Identifier 7Dh).
- Added a new element action, Add Entry Multiple (10b).
- Renamed existing element action Add/Update Entry to Add/Replace Entry.
- Added ten new Controller Metadata Element Types, 07h through 10h.
- Reorganized section 8.2 to reduce redundancy.
- Added a definition for Host Metadata Features.
- References:
  - NVMe-MI 1.1 sections 1 and 8.
  - Technical Proposal 6009.

Markup Conventions:

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

<Green Bracketed>: Notes to editor

1 Introduction

…

1.10 Definitions

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).

…

8 NVM Express Management Enhancements

This section describes NVMe Management Interface enhancements to the NVM Express specification.

8.1 Identify Controller

…

8.2 Management Interface Specific Features

The NVMe Get Features and Set Features Admin commands are used to retrieve and modify Feature values. Feature Identifiers 78h to 7Fh have been allocated by the NVM Express specification for this specification and are defined below.

Figure 1: NVMe Management Interface Feature Identifiers

Feature Identifier	NVMe Storage Device O/M [1]	NVMe Enclosure O/M [1]	Persistent Across Power States and Reset [2]	Uses Memory Buffer for Attributes	Description	Reference
78h to ~~7Dh~~ 7Ch	-	-	-	-	Reserved	
7Dh	M	M	No	Yes	Enhanced Controller Metadata	8.2.3
7Eh	M	M	No	Yes	Controller Metadata	8.2.4
7Fh	M	O	No	Yes	Namespace Metadata	8.2.5

NOTES:
1. O/M definition: O = Optional, M = Mandatory. Mandatory features shall be supported if the NVM Subsystem implements a Management Endpoint. These features are not mandatory if the NVM Subsystem does not implement a Management Endpoint.
2. This column is only valid if bit 4 in the Optional NVM Command Support field of the Identify Controller Data Structure is cleared to '0'. Refer to the NVM Express specification.

~~8.2.1 Controller Metadata~~

~~This feature is used to store metadata about the host platform in an NVM Subsystem for later retrieval. The values stored in the Controller Metadata Feature do not modify Controller behavior.~~

8.2.1 Persistent Event Log Interaction

The Persistent Event Log (refer to the NVM Express specification) defines an optional event type (Set Feature Event) which contains the data of a successful Set Features command if the Feature Identifier specified by that command is supported to be logged. Logging one or more of the NVMe Management Interface features (refer to Figure 137) is optional.

If logging of a Host Metadata Feature is supported, then the Persistent Event Log is able to contain information about the system environment in which the NVMe Storage Device is installed and which can be retrieved for diagnostic purposes.

The UTF-8 string associated with each Host Metadata Element Type is outside the scope of this specification. Because each Element Type is defined in this specification, diagnostic software used by different vendors to retrieve the Persistent Event Log can interpret the information across multiple systems and sites.

8.2.2 Host Metadata Overview

The Requester sends a Host Metadata data structure (refer to Figure 139) via the Set Features command specifying one of the Host Metadata Features (refer to section 1.8.TBD1). The Requester receives a Host Metadata data structure via the Get Features command specifying one of the Host Metadata Features.

The Host Metadata Features use the ~~Controller Metadata feature and the Namespace Metadata feature uses~~ NVMe Set ~~Feature~~ Features Command Dword 11 as shown in Figure 138.

Figure 138: ~~Host Metadata~~ Set Features – Command Dword 11

Bit	Description
31:15	Reserved

<table>
<tr><td></td><td>Element Action (EA): 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.

<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>00b</td><td>Add/~~Update~~ 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>~~10b to~~ 11b</td><td>Reserved</td></tr>
</table>

If the Element Action field is cleared to 00b (Add/~~Update~~Replace Entry) and a Metadata Element Descriptor with the specified Element Type (refer to Figure 140) does not exist in ~~Controller Metadata~~ the specified Host Metadata Feature value, then the Controller ~~creates a new~~ shall create the descriptor in the specified Host Metadata Feature value with the value in the ~~Controller~~ Host Metadata data structure. ~~This operation is performed in an atomic manner.~~

If the Element Action field is cleared to 00b~~h~~ (Add/~~Update~~Replace Entry) and ~~a~~ one Metadata Element Descriptor with the specified Element Type exists in the ~~Controller Metadata~~ specified Host Metadata Feature value, then the Controller ~~updates~~ shall replace the descriptor with the value in the ~~Controller~~ specified Host Metadata data structure.~~ This operation is performed in an atomic manner.~~

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.

~~If the Element Action field is set to 01b (Delete Entry) and a Metadata Element Descriptor with the specified Element Type does not exist in the Controller Metadata, then no operation is performed and the command completes successfully.~~

If the Element Action field is set to 01b (Delete Entry Multiple) ~~and a Metadata Element Descriptor with the specified Element Type exists in the Controller Metadata~~, then the Controller ~~deletes~~ shall delete all of the specified Metadata Element Descriptors from the specified Host Metadata Feature value, if any. ~~This operation is performed in an atomic manner.~~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.

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.

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 exist 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.

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.</td></tr>
<tr><td>14:13</td><td></td></tr>
<tr><td>12:00</td><td>Reserved</td></tr>
</table>

New Metadata Element Descriptors may be added, ~~updated~~ 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 ~~this Feature~~ a Host Metadata Feature, then a Host Metadata data structure, defined in Figure 139, 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 ~~update~~ replace a Metadata Element that causes the ~~stored Host Metadata data structure~~ Host Metadata Feature value of the specified feature to grow larger than 4 KiB, then the Controller shall abort the command with an Invalid Parameter Error Response. ~~The Host Metadata Data Structure for this feature is independent of the Host Metadata data structure for the Namespace Metadata feature described in section 8.2.2.~~

A Set Features command specifying one of the Host Metadata Features (refer to section 1.8.TBD1) does not affect the value of the other Host Metadata Features.

If a Get Features command is issued specifying one of the Host Metadata Features, all of the Metadata Element Descriptors present in the specified Host Metadata Feature value are added to a Host Metadata data structure (refer to Figure 139) and returned in the data buffer for that command. The data buffer size is equal to the size of the Host Metadata data structure and is 4 KiB in size.

Figure 2: Host Metadata Data Structure

Byte	Description
00	**Number of Metadata Element Descriptors:** This field contains the number of Metadata Element ~~Descriptors~~ descriptors in the data structure.
01	Reserved
x:02	**Metadata Element Descriptor 0:** This field contains the first Metadata Element descriptor.
y:x+1	**Metadata Element Descriptor 1:** This field contains the second Metadata Element descriptor or 0h if there is only 1 entry.
…	…
4095:z	**Metadata Element Descriptor N:** This field contains the (N+1)th Metadata Element descriptor or 0h if there are fewer than N+1 entries.

~~A~~ 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 ~~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 140.

The Number of Metadata Element Descriptors shall be cleared to 0h on a Controller Level Reset.

Figure 3: Metadata Element Descriptor

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

Figure 3: Metadata Element Descriptor

Bit	Description
07:06	Reserved
05:00	**Element Type (ET):** This field specifies the type of metadata stored in the descriptor. **Value** / **Definition** 00h: Reserved 01h to 017h: Element types defined by this specification. Controller Metadata Element types are defined in Figure 141. Namespace Metadata Element types are defined in Figure 142. 18h to 1Fh: Vendor Specific

8.2.3 Enhanced Controller Metadata (Feature Identifier 7Dh)

This feature is used to store metadata about the host platform in an NVM Subsystem for later retrieval. It is similar to the Controller Metadata Feature (refer to section 8.2.4), except that the Add Entry Multiple action is allowed for this feature.

The Enhanced Controller Metadata Feature is intended to supersede the Controller Metadata Feature. The Controller Metadata Feature is mandatory to provide backward compatibility with Management Controllers compliant to earlier versions of this specification.

If a Controller supports both the Enhanced Controller Metadata Feature and the Controller Metadata Feature, then the Requester should use the Enhanced Controller Metadata Feature and should not use the Controller Metadata Feature.

The metadata element types defined in Figure 141 are used by this feature.

8.2.4 Controller Metadata (Feature Identifier 7Eh)

This feature is used to store metadata about the host platform in an NVM Subsystem for later retrieval. It is similar to the Enhanced Controller Metadata Feature (refer to section 8.2.3), except that the Add Entry Multiple action is prohibited for this feature.

~~If a Get Features command is issued for this Feature, all Controller Metadata associated with the specified Controller is added to a Host Metadata Data Structure specified in Figure 139 and returned in the data buffer for that command. The data buffer size is equal to the size of the Host Metadata Data Structure and is 4,096 bytes in size.~~

The metadata element types defined in Figure 141 are used by this feature.

Figure 141: Controller Metadata Element Types

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

Value	Definition
09h	**Chipset Driver Version:** The chipset driver version as a UTF-8 string.
0Ah	**Operating System Name and Build:** The operating system name and build as a UTF-8 string.
0Bh	**System Product Name:** The system product name as a UTF-8 string.
0Ch	**Firmware Version:** The host firmware (e.g., UEFI) version as a UTF-8 string.
0Dh	**Operating System Driver Filename:** The operating system driver filename as a UTF-8 string.
0Eh	**Display Driver Name:** The display driver name as a UTF-8 string.
0Fh	**Display Driver Version:** The display driver version as a UTF-8 string.
10h	**Host-Determined Failure Record:** A failure record (e.g., the reason the host has flagged a failure for an NVMe Storage Device FRU which may be used for failure analysis) as a UTF-8 string.
~~07h~~ 11h to 17h	Reserved
18h to 1Fh	Vendor Specific

~~Controller Metadata is reset on a Controller Level Reset (i.e., the number of stored Metadata Element Descriptors is zero). Executing a Get Features command while the Controller is disabled returns zero Metadata Element Descriptors.~~

~~8.2.2 Namespace Metadata~~

8.2.5 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. ~~The values stored in the Namespace Metadata Feature do not modify Controller behavior on the namespace.~~ This feature is namespace specific. The Add Entry Multiple action is prohibited for this feature.

~~The Namespace Metadata feature uses Command Dword 11 as shown in Figure 138.~~

~~New Metadata Element Descriptors may be added, updated replaced, or deleted based on the action specified in the Element Action field.~~

~~If a Set Features command is submitted for this Feature, a Host Metadata data structure, defined in Figure 139, 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 update a Metadata Element Descriptor that causes the stored Host Metadata data structure to grow larger than 4 KiB, the Controller shall abort the command with an Invalid Parameter Error Response. The Host Metadata structure for this feature is independent of the Host Metadata data structure for the Controller Metadata feature described in section 8.2.1.~~

~~A Host Metadata data structure may contain up to one Metadata Element Descriptor of each element type. Each Metadata Element Descriptor contains the data structure shown in Figure 140.~~

~~If a Get Features command is issued for this Feature, all Namespace Metadata associated with the specified Controller is added to a Host Metadata Data Structure specified in Figure 139 and returned in the data buffer for that command. The data buffer size is equal to the size of the Host Metadata Data Structure and is 4 KiB in size.~~

~~Namespace Metadata is reset on a Controller Level Reset (i.e., the number of stored Metadata Element Descriptors is zero). Executing a Get Features command while the Controller is disabled returns zero Metadata Element Descriptors.~~

Figure 4: Namespace Metadata Element Types

Value	Definition
00h	Reserved

Figure 4: Namespace Metadata Element Types

Value	Definition
01h	**Operating System Namespace Name:** The name of the namespace in the operating system as a UTF-8 string.
02h	**Pre-boot Namespace Name:** The name of the namespace in the pre-boot environment as a UTF-8 string.
03h to 17h	Reserved
18h to 1Fh	Vendor Specific

9 Management Architecture

...

< End of changes >