



LEGAL NOTICE:

© **Copyright 2007 to 2022 NVM Express™, Inc. ALL RIGHTS RESERVED.**

This erratum 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 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 2022 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
c/o VTM, Inc.
3855 SW 153rd Drive
Beaverton, OR 97003
USA
info@nvmexpress.org

NVM Express™ Technical Errata

Errata ID	003
Revision Date	2022.01.10
Corrected Spec Ver.	NVMe Management Interface Specification, Revision 1.2a

Errata Author(s)

Name	Company
Mike Allison, Andres Baez, Myron Loewen, Peter Onufryk	Intel
Austin Bolen	Dell EMC
Mike Allison, Judy Brock	Samsung
Yoni Shternhell	WDC
John Geldman	Kioxia

Errata Overview

This ECN is based on NVMe Management Interface Specification, Revision 1.2a.

- Clarified that figure 2 is from the Requester's point of view.
- Clarified that Responders are allowed to process Command Messages in any order and Requester's must wait for a Response Message from a previous Request Message before sending another Request Message to guarantee ordering.
- Clarified the definition of Command Message.
- Clarified the definition of Request Message.
- Clarified how hosts often isolate SMBus/I2C channels.
- Updated the error handling to indicate Control Primitives report reasons for dropped MCTP traffic.
- Clarified that the use of MCTP is about message transport and not reliability.
- Removed the redundant MCTP Base Specification requirements in section 3.2.1.
- Removed the word "classified" in descriptive language to be the actual message type.
- Removed text that was redundant in a sub-section to the parent section in section 4.1.
- Clarified that a Response Message that is not an error includes the More Processing Required Response.
- Remove the repeated requirement from MCTP specifications that Control Primitives consist of exactly one MCTP packet.
- Indented all Command Servicing State descriptions to disassociate the text from text associates with a Command Message processing.
- Clarified the Tag field definition.
- Removed repeated statement detailing responses transmitted at the completion of a Control Primitive.
- Clarified that packet times are associated with the MCTP Base Specification.
- Clarified the Command Slot number on the Pause Flag Status Slot 0 & 1 fields.
- Clarified the packet to be transmitted on a Resume Control Primitive.
- Clarified the definition of the Clear Error State Flags (CESF).
- Clarified that a Replay Control Primitive to resume from a paused condition.
- Moved text around to keep the order of presenting Command Messages prior to Response Messages.
- Clarified that a Controller Health Status poll is able to request for controller type that do not exist without an error.
- Updated the Critical Warning field to match NVM Express Base Specification to include the Persistent Memory Region Error bit.
- Clarified that the I/O Command Set Identifier field is valid if the I/O Command Set is not enabled.
- Changed the Error Response of a prohibited NVMe Admin Command to an Invalid Command Opcode Error Response.
- Indicated that the value of FFh is not allowed in the Number of Ports field (a 0's based number) in the NVM Subsystem Information data structure as the port identifier of 256 is not allowed to be reported.
- Clarified that the PCIe Maximum Payload Size field in Figure 96 has PCIe restrictions for ARI devices and Non-ARI Multi-Function devices.
- Clarified that each Management Endpoint is allowed to support optional log pages and features independent of the other Management Endpoints.
- Log pages supported and Features supported have the same requirements for an SMBus/I2C and PCIe VDM Management Endpoints so combined the requirements to simplify Figure 122 and Figure 126.
- Clarified restrictions in the NVMe MultiRecord Area and NVMe PCIe Port MultiRecord Area by using nested lists.
- Updated fixed values to state a required value in many subsections of section 8.2.
- Clarified that a NVM Subsystem Reset may occur with a PCIe Memory Write command.

Revision History

Revision Date	Author	Change Description
2020.09.17	Mike Allison	<ul style="list-style-type: none">• Initial draft.• Updated the PCIe Maximum Payload Size field.
2021.04.02	Austin Bolen	<ul style="list-style-type: none">• Updated to new ECN template.
2021.07.08	Mike Allison	<ul style="list-style-type: none">• Renumbered to ECN 003 using global numbers as defined by the Technical WG.

		<ul style="list-style-type: none"> Applies the postponed changes from the NVMe-MI 1.2 member review.
2021.07.19	Mike Allison	<ul style="list-style-type: none"> Changes due to review during NVMe-MI Task Group meeting.
2021.07.22	Mike Allison	<ul style="list-style-type: none"> Changed “whether” to “if” on conditional text.
2021.07.26	Mike Allison	<ul style="list-style-type: none"> Incorporated Myron’s feedback. Updates due to comment review during Management Interface Task Group meeting. Added more “whether” statements to replace with “if” per Task Group Request. Added figure 26 to indicate ranges of status value that indicate or do not indicate errors. Removed closed comments. Changed “NVMe Base Specification” to “NVM Express Base Specification” Changed “NVMe Subsystem” to “NVM Subsystem”
2021.07.27	Mike Allison	<p>Fixed the following identified by Yoni Shternhell:</p> <ul style="list-style-type: none"> Fixed “in band” Fixed “NVMe Subsystem” Fixed “NVMe Express Base Specification” Fixed “NVMe Admin command”
2021.08.02	Mike Allison	<ul style="list-style-type: none"> Updates during review in the NVMe-MI Task Group meeting.
2021.08.02	Mike Allison	<ul style="list-style-type: none"> Removed closed comments. Updated figure 40 with actual requirement instead of generic statement. Created a list where the references are in different specifications for clarity.
2021.08.09	Mike Allison	<ul style="list-style-type: none"> Removing the changes of “whether” to “if” – later ECN if desired. Added missing title change from TP 4047b Added mandatory requirements accidentally changed by TP
2021.08.16	Mike Allison	<ul style="list-style-type: none"> Updates during NVMe-MI review.
2021.08.23	Mike Allison	<ul style="list-style-type: none"> Updates during NVMe-MI review.
2021.08.28	Mike Allison	<ul style="list-style-type: none"> Added the requirements for log pages and features to remove redundant columns.
2021.08.20	Mike Allison	<ul style="list-style-type: none"> Editorial changes made during NVMe-MI review.
2021.08.31	Mike Allison	<ul style="list-style-type: none"> Indented the Command Slot state text to differentiate paragraphs associated with the Transmit state from paragraph associated with the section. Added clarifying text from TP 6022 to indicate that each Management Endpoint has unique optional support for log pages and Feature Identifiers. TP 6022 updated the NVMe Base Specification and not the NVMe-MI specification. Reject the removal of the text in PCIe Max Payload field that is just stating a PCIe fact.
2021.09.10	Mike Allison	<ul style="list-style-type: none"> Added Judy Brock’s update to change “applicable” to “supported” for the NVM Express Admin Command Set in the out-of-band mechanism.
2021.09.13	Mike Allison	<ul style="list-style-type: none"> Removed text fixed in NVMe-MI 1.2a and aligned to new figure numbers. Added “or” to nested list in section 8.23 and 8.2.4. Updates due to review during Management Interface Task Group meeting.
2021.09.20	Mike Allison	<ul style="list-style-type: none"> Removed comments. Accepted all changes.
2021.09.27	Mike Allison	<ul style="list-style-type: none"> Incorporated Austin Bolen’s comments – many editorial changes.
2021.10.25	Mike Allison	<ul style="list-style-type: none"> Incorporated Gerry Houlder’s 30-day member review comments.
2021.11.4	Mike Allison	<ul style="list-style-type: none"> Accepted all changes and removed all comments for starting integration.
2021.11.17	Mike Allison	<ul style="list-style-type: none"> Added missing black text to align to specification.
2021.12.06	Devin Allison	<ul style="list-style-type: none"> Integrated
2021.12.07	Mike Allison	<ul style="list-style-type: none"> Converted blue text to black
2021.12.12	Devin Allison	<ul style="list-style-type: none"> Added Austin Bolen’s update to change “as shown below.” to “as follows:” in section 4.1.1

2021.12.16	Mike Allison	<ul style="list-style-type: none"> Fixed editorial changes identified by Paul Suhler. Updated the copyright dates since not ratified until 2022.
2021.12.17	Mike Allison	<ul style="list-style-type: none"> Added the high level section numbers for the navigation.
2022.01.10	Mike Allison	<ul style="list-style-type: none"> Updated navigation pane.

Description for Changes Document

Feature Enhancements:

- PCIe Max Payload Size updates
 - **New requirement / incompatible change**
 - Clarified the requirements for the value that is reported in the PCIe Max Payload Size field for ARI Devices and Non-ARI Multi-Function Devices.
- Status Flags field of the Subsystem Management Data Structure
 - **New requirement / incompatible change**
 - Clarified that both the bits 1:0 in the field shall be set to 11b. Previous wording was not clear if both bits are to be set to '1'.
- Error Response for a prohibited NVMe Admin Command
 - **New requirement / incompatible change**
 - The processing of a prohibited NVMe Admin Command returns an Invalid Command Opcode Error Response as opposed to the Parameter Error Response.
- Number of Ports field in the NVM Subsystem Information data structure unsupported value
 - **New requirement / incompatible change**
 - The value of FFh is not supported as a port identifier of 256 is not allowed to be reported.

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

Description of Specification Changes

Modify Section 1.3.1.1 as follows:

1 Introduction

...

1.3 Theory of Operation

...

1.3.1 Out-of-Band Theory of Operation

...

1.3.1.1 Management Component Transport Protocol

The out-of-band mechanism utilizes the Management Component Transport Protocol (MCTP) as the transport and utilizes existing MCTP SMBus/I2C and PCIe bindings for the physical layer. Command Messages are submitted to one of two Command Slots associated with a Management Endpoint contained in an NVM Subsystem. Figure 2 shows the NVMe-MI out-of-band protocol layering [from the Requester's point of view](#).

Modify Section 1.4 as follows:

1.4 NVM Subsystem Architectural Model

...

Each NVMe Controller in the NVM Subsystem shall provide an NVMe Controller Management Interface (hereafter referred to as simply Controller Management Interface). The Controller Management Interface processes Controller operations on behalf of any Controller (in-band tunneling mechanism) or Management Endpoint (out-of-band mechanism) in the NVM Subsystem. [NVMe](#) Controllers or Management Endpoints may route commands to any NVMe Controller in the NVM Subsystem. A Controller Management Interface logically processes one operation at a time. A Controller Management Interface is not precluded from processing two or more operations in parallel; however, there shall always be an equivalent pattern of sequential operations with the same results. [Responders are permitted to process Command Messages in any order. If the Requester requires Command Messages to be processed in a particular order, then the Requester waits for the Response Message of one Command Message before sending the next Command Message.](#)

Modify Section 1.8.2 as follows:

1.8 Definitions

...

1.8.2 Command Message

A ~~type of~~ Request Message that contains an NVMe Admin Command, PCIe Command, or NVMe-MI Command.

Modify Section 1.8.25 as follows:

1.8.25 Request Message

An NVMe-MI Message originating from a Requester. A Request Message may be a Command Message or a Control Primitive. ~~Request Messages may be used in both the out-of-band mechanism and the in-band tunneling mechanism.~~

Modify Section 2.2 as follows:

2 Physical Layer

...

2.2 SMBus/I2C

...

Host platforms expecting to be used with one or more Management Endpoints (e.g., data center platforms and workstations) ~~often should~~ isolate SMBus/I2C channels to avoid a Management Endpoint conflicting with the address of another SMBus/I2C element. An SMBus/I2C address conflict may occur when a Management Endpoint that does not support ARP is used with platforms that do not isolate SMBus/I2C channels (e.g., some client platforms). ARP ~~is may be~~ used to dynamically reassign SMBus/I2C addresses in a system when supported by both the Management Controller and the NVMe Storage Devices or NVMe Enclosure.

...

Modify Section 2.3 as follows:

2.3 Error Handling

Physical layer errors are handled as specified by the corresponding physical layer specification and MCTP transport binding specification. ~~This specification does not require any physical layer specific error handling requirements beyond those outlined in the MCTP transport binding specifications. This specification defines Control Primitives that add the capability of reporting reasons for dropped MCTP traffic (refer to section 4.2.1.4) and the capability of replaying dropped MCTP traffic (refer to section 4.2.1.5).~~

Modify Section 3.2 as follows:

3 Message Transport

...

3.2 Out-of-Band Message Transport

The out-of-band mechanism defined in this specification utilizes MCTP ~~for as a reliable~~ in-order message transport between a Management Controller and a Management Endpoint.

~~This section summarizes the NVMe-MI MCTP packet format.~~ A Management Endpoint compliant to this specification shall implement all required behaviors detailed in the Management Component

Transport Protocol (MCTP) Base Specification and corresponding MCTP transport binding specification in addition to the requirements outlined in this specification (e.g., the Message Integrity Check algorithm).

Modify Section 3.2.1 as follows:

3.2.1 MCTP Packet

...

~~A compliant Management Endpoint shall implement all MCTP required features defined in the MCTP Base Specification. Optional features may be supported.~~

Modify Section 4.1 as follows:

4 Message Servicing Model

...

4.1 NVMe-MI Messages

Figure 24 illustrates the taxonomy of NVMe-MI Messages. The two main categories of NVMe-MI Messages are Request Messages and Response Messages. Request Messages are sent by a Management Controller to a Management Endpoint when using the out-of-band mechanism. Request Messages are sent by host software to an NVMe Controller when using the in-band tunneling mechanism. The entity sending the Request Message is collectively referred to as the Requester and the entity receiving the Request Message is collectively referred to as the Responder. After receiving a Request Message, the Responder processes the Request Message. When processing is complete, the Responder sends a Response Message back to the Requester.

A Request Message ~~may be classified as~~ is a Command Message or a Control Primitive. A Command Message ~~s specifies~~ specifies an operation to be performed by the Responder and ~~may be further classified as~~ is an NVMe-MI Command, an NVMe Admin Command, or a PCIe Command. Control Primitives are used in the out-of-band mechanism to affect the servicing of a previously issued Command Message or get the state of a Command Slot and Management Endpoint (refer to section 4.2.1).

A Response Message ~~may be classified as~~ is a Success Response or an Error Response.

Modify Section 4.1.1 as follows:

4.1.1 Request Messages

~~Request Messages are NVMe-MI Messages that are generated by a Requester to send to a Responder.~~

Request Messages specify an action to be performed by the Responder. ~~Request Messages are either Control Primitives (refer to section 4.2.1) or Command Messages. The format of the Message Body for a Command Message is command set specific and is specified by the NMIMT field in the Message Header. The NMIMT field specifies the Request Message type. The format of the Message Body is determined by the Request Message types as follows:~~

~~The NVMe Management Interface supports three command sets:~~

- Control Primitives (refer to section 4.2.1);
- The Management Interface Command Set ~~as described in~~ (refer to section 5);
- The NVMe Express Admin Command Set ~~as described in~~ (refer to section 6); and
- The PCIe Command Set ~~as described in~~ (refer to section 7).

...

Modify Section 4.1.2 as follows:

4.1.2 Response Messages

...

Response Message Status values are summarized in Figure 27. A Response Message Status of Success indicates that the corresponding Request Message completed successfully and that the Response Message is a Success Response. The format of the Response Body for a Success

Response is dependent on the NVMe-MI Message Type (refer to Figure 19) and is described in the section defining each NVMe-MI Message Type later in this specification.

A Response Message Status other than Success indicates that:

- ~~that~~ an error occurred during servicing of the corresponding Request Message and that the Response Message is an Error Response; or
- more time is required for the processing of the corresponding Request Message and that the Response Message is a More Processing Required Response.

The format of the Response Body is dependent on the Response Message Status. Figure 27 references the section that defines the format of the Response Message for each Response Message Status value. If multiple errors are present, a Responder may choose which error status to report.

Figure 27: Response Message Status Values

Value	Description	Response Message Format Section
Status Values that do not indicate an error (i.e., Success Response).		
00h	Success: The command completed successfully.	4.1.2.1
01h	More Processing Required: The Command Message is in progress and requires more time to complete processing. When this Response Message Status is used in a Response Message, a subsequent Response Message contains the result of the Command Message. This Response Message Status shall not be sent more than once per Command Message, except for retransmission due to a Replay Control Primitive as described in section 4.2.1.5.	4.1.2.3
Status Values that indicate an error (i.e., Error Response).		
02h	Internal Error: The Request Message could not be processed due to a vendor specific internal error.	4.1.2.1
03h	Invalid Command Opcode: The associated command opcode field is not valid. Invalid opcodes include reserved and optional opcodes that are not implemented.	4.1.2.1
04h	Invalid Parameter: Invalid parameter field value. Request Messages received with reserved or unimplemented values in defined fields shall be completed with an Invalid Parameter Error Response. Other error conditions that result in Invalid Parameter Error Response are specified elsewhere in this specification.	4.1.2.2
05h	Invalid Command Size: The size of the Message Body of the Request Message was different than expected due to a reason other than too much or too little Request Data (e.g., the Request Message did not contain all the required parameters or Request Data was present when not expected). The expected size of the Message Body is determined by the NVMe-MI Message Type and opcode assuming no other errors are detected (e.g., Invalid Command Opcode or Invalid Parameter).	4.1.2.1
06h	Invalid Command Input Data Size: The Command Message requires Request Data and contains too much or too little Request Data.	4.1.2.1
07h	Access Denied: A Request Message was prohibited from being processed due to a vendor specific protection mechanism or the Command and Feature Lockdown feature (refer to the NVM Express Base Specification).	4.1.2.1
08h to 1Fh	Reserved	-
20h	VPD Updates Exceeded: More updates to the VPD are attempted than allowed.	4.1.2.1
21h	PCIe Inaccessible: The PCIe functionality is not available at this time.	4.1.2.1
22h	Management Endpoint Buffer Cleared Due to Sanitize: An attempt was made to read data as defined in section 4.2.3 in the Management Endpoint Buffer that was zeroed due to a sanitize operation.	4.1.2.1
23h	Enclosure Services Failure: The Enclosure Services Process has failed in an unknown manner.	4.1.2.1
24h	Enclosure Services Transfer Failure: Communication with the Enclosure Services Process has failed.	4.1.2.1

Figure 27: Response Message Status Values

Value	Description	Response Message Format Section
25h	Enclosure Failure: An unrecoverable enclosure failure has been detected by the Enclosure Services Process.	4.1.2.1
26h	Enclosure Services Transfer Refused: The NVM Subsystem or Enclosure Services Process indicated an error or an invalid format in communication.	4.1.2.1
27h	Unsupported Enclosure Function: An SES Send command has been attempted to a simple Subenclosure.	4.1.2.1
28h	Enclosure Services Unavailable: The NVM Subsystem or Enclosure Services Process has encountered an error but may become available again.	4.1.2.1
29h	Enclosure Degraded: A noncritical failure has been detected by the Enclosure Services Process.	4.1.2.1
2Ah	Sanitize In Progress: The requested command is prohibited while a sanitize operation is in progress. Refer to section 8.1.	4.1.2.1
2Bh to DFh	Reserved	-
Status Values that may or may not indicate an error.		
E0h to FFh	Vendor Specific	Vendor Specific

...

Modify Section 4.2 as follows:

4.2 Out-of-Band Message Servicing Model

The out-of-band mechanism in this specification utilizes a request and response servicing model. A Management Controller sends a Request Message to a Management Endpoint, the Management Endpoint processes the Request Message, and when processing has completed, sends a Response Message back the Management Controller. Under no circumstances does a Management Endpoint generate an unsolicited Response Message (i.e., a Response Message that does not correspond to a previously received Request Message).

~~Unlike other NVMe-MI Messages that may span multiple MCTP packets, NVMe-MI Messages containing a Control Primitive shall consist of exactly one MCTP packet.~~

This specification utilizes Command Slots for Command Message servicing. A Management Controller should not send a new Command Message to a Command Slot until the Response Message for the previously issued Command Message to that Command Slot has been received. Each Management Endpoint contains two Command Slots that each include state information and a Pause flag (refer to section 4.2.1.4).

...

<Editor – No need to number the states. Remove numbering but keep indentation>

4. **Idle:** This is the default state of the command servicing state machine (e.g., following a reset). Command servicing transitions from Idle to the Receive state when the first MCTP packet of a Command Message is received (i.e., an MCTP packet with the SOM bit in the MCTP packet header set to '1' and the Message Type set to 4h).

- 2- **Receive:** The state when the first packet of a Command Message has been received and the Command Message is being assembled or validated. Command servicing transitions from Receive to the Idle state when an Abort Control Primitive is received, an error is detected in message assembly (refer to section 3.2.1.1), or the Message Integrity Check fails (refer to section 3.1.1.1). Command servicing transitions from Receive state to the Process state when a Command Message is assembled and the message integrity check is successful.
- 3- **Process:** The state when a Command Message is processed. Processing of a Command Message consists of checking for errors with the Command Message and performing the actions specified by the Command Message or aborting the Command Message. Command servicing transitions from Process to the Transmit state when a Response Message is required to be sent (i.e., processing of the Command Message has completed or command processing is expected to exceed the corresponding MCTP transport binding specification response timeout). Command servicing transitions from the Process state to the Idle state due to an Abort Control Primitive (refer to section 4.2.1.3).
- 4- **Transmit:** The state in which a Response Message for the Command Message is transmitted to the Management Controller. Command servicing transitions from the Transmit to the Idle state once the entire NVMe-MI Message associated with the response to the Command Message has been transmitted on the physical medium or due to an Abort Control Primitive (refer to section 4.2.1.3).

If, and only if, both the command servicing did not complete in the Process state and the Command Slot is not paused, then the Management Endpoint transmits a Response Message with status More Processing Required. If the Command Message requires more processing, then the Command Slot shall transition back to the Process state.

Modify Figure 35 and section 4.2.1 as follows:

4.2.1 Control Primitives

...

Figure 35: Control Primitive Fields

Bytes	Description
...	
05	Tag (TAG): This field contains an opaque value that is sent from the Management Controller in the Control Primitive and returned by the Management Endpoint in the associated Response Message. A Management Controller is allowed to use any value in this field.
...	

...

~~A Management Endpoint transmits a Response Message to the Management Controller when the actions associated with that Control Primitive have completed.~~

Modify section 4.2.1.1 as follows:

4.2.1.1 Pause

The Pause Control Primitive is used to suspend response transmission and suspend the timeout waiting for packet, [as defined in the MCTP Base Specification](#), for both Command Slots in a Management Endpoint. The CSI bit in a Pause Control Primitive is not used and shall be cleared to 0h. If the CSI bit is set to '1', then the Management Endpoint should transmit an Invalid Parameter Error Response with the PEL field indicating the CSI bit.

...

Figure 39: Pause Control Primitive Success Response Fields

Bytes	Description	
07:06	Control Primitive Specific Response (CPSR): This field is used to return Control Primitive specific status.	
	Bits	Description
	15:02	Reserved
	01	Pause Flag Status Slot 1 (PFSS1): This bit indicates whether or not Command Slot 1 is paused after completing the Pause Control Primitive. This bit set to '1' indicates that Command Slot 1 is paused. This bit cleared to '0' indicates that Command Slot 1 is not paused.
	00	Pause Flag Status Slot 0 (PFSS0): This bit indicates whether or not Command Slot 0 is paused after completing the Pause Control Primitive. This bit set to '1' indicates that Command Slot 0 is paused. This bit cleared to '0' indicates that Command Slot 0 is not paused.

...

<Editor – Indenting the text associated with the state descriptions.>

The result of a Pause Control Primitive on a Command Slot is dependent on the command servicing state of the Command Slot when the Pause Control Primitive is received, as described below:

Idle: The Pause Control Primitive has no effect, and the Pause Flag is not changed (i.e., remains cleared to '0'). Refer to section 4.2.1.4.

Receive: The Pause Control Primitive sets the Pause Flag to '1' (refer to section 4.2.1.4) and alerts the Management Endpoint that remaining MCTP packets associated with the Command Message may be delayed. Further packets sent to this Command Slot while the Pause Flag is set to '1' are received normally.

Process: The Pause Control Primitive sets the Pause Flag to '1' (refer to section 4.2.1.4). The Pause Flag has no effect on the command processing in the Command Slot. Upon completion of command processing, the Command Slot shall transition to the Transmit state.

Transmit: The Pause Control Primitive sets the Pause Flag to '1' (refer to section 4.2.1.4) suspending transmission of Response Messages on a packet boundary. The Management Endpoint should pause transmission as soon as possible after receiving a Pause Control Primitive.

The Management Endpoint shall transmit a Response Message with success status after receiving the Pause Control Primitive. It is not an error to issue a Pause Control Primitive when a Command Slot is already paused.

While the Pause Flag is set to '1', the Management Endpoint disables the timeout waiting for packet, as defined in the MCTP Base Specification, timer and does not transmit Response Messages to Command Messages. The timeout waiting for a packet is the lesser of 100 ms or the time defined in the appropriate MCTP transport binding specification. The Management Controller should not send Command Messages to a Command Slot that is paused.

Modify section 4.2.1.2 as follows:

4.2.1.2 Resume

The Resume Control Primitive is used to resume from a paused condition. This is the complement to the Pause Control Primitive.

Like the Pause Control Primitive, the Resume Control Primitive affects both slots and the CSI bit in a Resume Control Primitive shall be cleared to '0'. If a Command Slot was not paused before receiving the Resume Control Primitive, the Resume Control Primitive completes successfully and has no effect.

Note that the Resume Control Primitive causes a Management Endpoint to transmit the packet after the last packet the Management Endpoint transmitted prior to being paused. If the last packet transmitted was not received by the Management Controller, then, after the first packet of the resumed Response Message is transmitted by the Management Endpoint, the Management Controller should detect an out-of-sequence packet sequence number in the resumed Response Message and drop the Response Message. To avoid this synchronization issue, the Management Controller should issue a Replay Controller Primitive specifying the packet number in the Response Replay Offset field from which the Response Message is replayed.

The CPSP field for the Resume Control Primitive is reserved. The CPSR field in the Control Primitive Success Response is reserved.

<Editor – Indenting the text associated with the state descriptions.>

The result of a Resume Control Primitive is based on the state of a Command Slot when the Resume Control Primitive is received, as described below:

Idle: The Resume Control Primitive has no effect.

Receive: The Resume Control Primitive alerts the Management Endpoint that transmission of any remaining MCTP packets associated with the Command Message is resuming. The Pause Flag is cleared to '0' (refer to section 4.2.1.4).

Process: If the Command Slot is paused and a More Processing Required Response has not yet been transmitted for the Command Message being processed, then the request-to-response timer shall be reset and restarted (refer to section 4.2.2.1 for details on the request-to-response time). The Pause Flag is cleared to '0' (refer to section 4.2.1.4).

Transmit: The Management Endpoint resumes transmission of the Response Message corresponding to the Command Message associated with the Command Slot after responding to the Resume Control Primitive. The Pause Flag is cleared to '0' (refer to section 4.2.1.4).

The Management Endpoint shall transmit a Control Primitive Response Message with success status after receiving the Resume Control Primitive.

Modify section 4.2.1.3 as follows:

4.2.1.3 Abort

...

<Editor – Indenting the text associated with the state descriptions.>

The result of an Abort Control Primitive is based on the command servicing state of the specified Command Slot when the Abort Control Primitive is received, as described below:

Idle: The Abort Control Primitive has no effect. The Management Endpoint shall transmit a Response Message with success status and the CPAS field cleared to 0h.

Receive: The Management Endpoint discards the contents of the Command Slot and transitions to the Idle state. The Management Endpoint shall transmit a Response Message with success status and the CPAS field set to 1h.

Process: The Abort Control Primitive causes processing of the command in the Command Slot to be aborted:

- a) ~~if~~ the Abort Control Primitive was received before command processing started, the Management Endpoint discards the contents of the Command Slot and transitions to the Idle state. The Management Endpoint shall transmit a Success Response and the CPAS field set to 1h; or
- b) ~~if~~ the Abort Control Primitive was received while the command is being processed, the Management Endpoint discards the contents of the Command Slot and transitions to the Idle state. The Management Endpoint attempts to abort the command:
 - ~~if~~ the command is aborted and had no effect on the NVM Subsystem, then the Management Endpoint shall transmit a Success Response and the CPAS field set to 1h; or
 - ~~if~~ the Management Endpoint is not able to abort the command, then the Management Endpoint shall transmit a Success Response and set the CPAS field to 2h.

Transmit: The Management Endpoint discards the contents of the Command Slot and transitions to the Idle state. The Management Endpoint transmits a Response Message with success status and the CPAS field cleared to 0h.

It is not a Management Endpoint error if the Management Controller issues an Abort Control Primitive to a Command Slot that is paused. The state of Command Slot is reinitialized clearing the Pause Flag to '0'.

Modify section 4.2.1.4 as follows:

4.2.1.4 Get State

...

Figure 41: Get State Control Primitive Request Message Fields

Bytes	Description	
07:06	Control Primitive Specific Parameter (CPSP): This field is used to pass Control Primitive specific parameter information.	
	Bits	Description
	15:01	Reserved
	00	Clear Error State Flags (CESF): This bit specifies whether or not to clear the error state flags when completing this command. If this bit is set to '1', then the Management Endpoint shall clear the error state flags. If this bit is cleared to '0', then the Management Endpoint shall not clear the error state flags.

...

Modify section 4.2.1.5 as follows:

4.2.1.5 Replay

The Replay Control Primitive is used to retransmit the Response Message for the last Command Message processed in a Command Slot and causes the Pause Flag for each Command Slot to be cleared to '0'.

The replayed Response Message forms a new MCTP Response Message with Message Data starting from Response Replay Offset of the original Response Message and continuing to the end of the Response Message, including the original MIC. The first packet shall have SOM set to '1' and shall include the Message Header of the original Response Message even if the Response Replay Offset is not 0h. The Msg tag in each packet of the replayed Response Message shall be set to the value of the Msg tag in the associated Replay Control Primitive. Refer to the MCTP Base Specification for the definition of the Msg tag.

...

<Editor – Indenting the text associated with the state descriptions.>

The result of a Replay Control Primitive is based on the command servicing state of the specified Command Slot when the Replay Control Primitive is received, as described below:

Idle: The Replay Control Primitive requests retransmission of the completion at the offset specified by the RRO field if such a completion is available:

- a) If the Replay Control Primitive was received following an Abort Control Primitive or a reset (refer to section 8.3) before any Command Messages are processed, then there is no Response Message available to retransmit. The Management Endpoint shall transmit a Response Message with success status with the RR bit cleared to '0'; or
- b) If the Replay Control Primitive was received following the processing of one or more Command Messages, then the Management Endpoint shall transmit a Response Message with success status with the RR bit set to '1'. The Management Endpoint transmits the MCTP packets associated with the requested Response Message after the Control Primitive Success Response.

Receive: The Management Endpoint transmits a Response Message with success status with the RR bit cleared to '0'.

Process: If a More Processing Required Response has not been transmitted for the Command Message being processed, then a Success Response shall be transmitted with the RR bit cleared to '0'.

If a More Processing Required Response has been transmitted, then a Success Response shall be transmitted with the RR bit set to '1' and then the More Processing Required Response shall be retransmitted. The Management Endpoint shall update the More Processing Required Time field in the Response Message with the current worst-case amount of additional time that the Management Controller should wait for the Management Endpoint to complete processing of the Command Message.

Transmit: The Management Endpoint stops transmitting response packets for the Command Slot and then transmits a Response Message with success status with the RR bit set to '1'. The Management Endpoint transmits a Response Message containing the packets starting at the packet offset specified in the Response Replay Offset field of the Replay Control Primitive after

the Control Primitive Success Response. The Command Slot remains in the Transmit state until retransmission is complete.

It is not an error to issue a Replay Control Primitive to a Command Slot that is paused. A ~~response~~ **Response Message** is transmitted even if the Command Slot is paused at any time during the response, including before the first packet was transmitted. After successful completion of the Replay Control Primitive, neither Command Slot is paused (i.e., there is an implicit Resume Control Primitive affecting both Command Slots when processing the Replay Control Primitive **except that the Management Endpoint shall not transmit a Response Message**).

Modify section 4.3 as follows:

4.3 In-Band Tunneling Message Servicing Model

The in-band tunneling mechanism in this specification utilizes two NVMe Admin Commands (NVMe-MI Send and NVMe-MI Receive). ~~The NVMe-MI Send command is used to tunnel an NVMe-MI Command from host software to an NVMe Controller that transfers data from the host to the NVMe Controller (similar to a write operation) or to instruct the Responder to perform an action (e.g., to reset the NVM Subsystem using the Reset command). The NVMe-MI Receive command is used to tunnel an NVMe-MI Command from a host to an NVMe Controller that transfers data from the NVMe Controller to the host (similar to a read operation).~~ Figure 60 specifies whether an NVMe-MI Command is tunneled via the NVMe-MI Send command or the NVMe-MI Receive command.

~~Refer to the NVM Express Base Specification for additional details on the NVMe-MI Send and NVMe-MI Receive commands. Additional details on NVMe-MI Send are in section 4.3.1 and additional details on NVMe-MI Receive are in section 4.3.2.~~

NVMe-MI Commands may apply to the NVM Subsystem, Controllers, and/or Namespaces. If a tunneled NVMe-MI Command applies to one or more Controllers, then the applicable Controller(s) are specified by fields in the tunneled NVMe-MI Command. Note that unlike some other NVMe Admin Commands, the Controller to which the tunneled NVMe-MI Command is issued is not used to determine which Controller the tunneled NVMe-MI Command applies to. If the tunneled NVMe-MI Command applies to one or more Namespaces, then the applicable Namespace(s) are specified by fields in the tunneled NVMe-MI Command. Note that the Namespace Identifier (NSID) field of the tunneled NVMe-MI Command (bytes 7:4 of the Submission Queue Entry) is not used and should be cleared to 0h by host software.

For details on the NVMe-MI Send command refer to:

- section 4.3.1; and
- the NVM Express Base Specification.

For details on the NVMe-MI Receive command refer to:

- section 4.3.2; and
- the NVM Express Base Specification.

Modify section 4.3.1 as follows:

4.3.1 NVMe-MI Send Command

The NVMe-MI Send command is an NVMe Admin Command as defined by this specification and the NVM Express Base Specification. It is used to tunnel an NVMe-MI Command in-band from host software to an NVMe Controller that transfers data from a host to an NVMe Controller (similar to a

write operation) or to instruct the Responder to perform an action (e.g., to reset the NVM Subsystem using the Reset command). The data being transferred or action to be performed is in one or more of the following locations: Request Data, NVMe Management Dword 0, NVMe Management Dword 1. Figure 60 specifies which NVMe-MI Commands are tunneled via the NVMe-MI Send command.

~~NVMe-MI Commands may apply to the NVM Subsystem, Controllers, and/or Namespaces. If the tunneled NVMe-MI Command applies to one or more Controllers, then the applicable Controller(s) are specified by fields in the tunneled NVMe-MI Command. Note that unlike some other Admin Commands, the Controller to which the NVMe-MI Send command is issued is not used to determine which Controller the tunneled NVMe-MI Command applies to. If the tunneled NVMe-MI Command applies to one or more Namespaces, then the applicable Namespace(s) are specified by fields in the tunneled NVMe-MI Command. Note that the Namespace Identifier (NSID) field of the NVMe-MI Send command (bytes 7:4 of the Submission Queue Entry) is not used and should be cleared to 0h by host software.~~

~~The mapping of how an NVMe-MI Command is tunneled inside of NVMe-MI Send commands is described in section 4.3.1.1. The NVMe-MI Send command servicing model is described in section 4.3.1.2.~~

Modify section 4.3.2 as follows:

4.3.2 NVMe-MI Receive Command

The NVMe-MI Receive command is an NVMe Admin Command as defined by this specification and the NVM Express Base Specification. It is used to tunnel an NVMe-MI Command in-band from host software to an NVMe Controller that transfers data from an NVMe Controller to a host (similar to a read operation). The data being transferred is in one or more of the following locations: Response Data, NVMe Management Response. Figure 60 specifies which NVMe-MI Commands are tunneled via the NVMe-MI Receive command.

~~NVMe-MI Commands may apply to the NVM Subsystem, Controllers, and/or Namespaces. If the tunneled NVMe-MI Command applies to one or more Controllers, then the applicable Controller(s) are specified by fields in the tunneled NVMe-MI Command. Note that unlike some other Admin Commands, the Controller to which the NVMe-MI Receive command is issued is not used to determine which Controller the tunneled NVMe-MI Command applies to. If the tunneled NVMe-MI Command applies to one or more Namespaces, then the applicable Namespace(s) are specified by fields in the tunneled NVMe-MI Command. Note that the Namespace Identifier (NSID) field of the NVMe-MI Receive command (bytes 7:4 of the Submission Queue Entry) is not used and should be cleared to 0h by host software.~~

~~The mapping of how an NVMe-MI Command is tunneled inside of an NVMe-MI Receive command is described in section 4.3.2.1. The NVMe-MI Receive command servicing model is described in section 4.3.2.2.~~

Modify section 5 as follows:

5 Management Interface Command Set

...

The NVMe-MI Message structure with **all** fields that are common to all NVMe-MI Messages is defined in section 3.1. ~~The Response Message structure for the Management Interface Command Set is~~

~~defined in section 4.1.2.~~ The Message Body for the Management Interface Command Set is shown in Figure 56 and Figure 57. Command specific fields for the Management Interface Command Set are defined in this section. [The Response Message structure for the Management Interface Command Set is defined in section 4.1.2.](#)

...

Modify section 5.3 as follows:

5.3 Controller Health Status Poll

...

Figure 77: Controller Health Status Poll – NVMe Management Dword 0

Bits	Description
...	
30:27	Reserved
26	Include SR-IOV Virtual Functions (INCVF): When this bit is set to '1', a Controller Health Data Structure is returned for Controllers associated with SR-IOV Virtual Functions (VFs). It is not an error if this bit is set to '1' and SR-IOV Virtual functions do not exist.
25	Include SR-IOV Physical Functions (INCPF): When this bit is set to '1', a Controller Health Data Structure is returned for Controllers associated with SR-IOV Physical Functions (PFs). It is not an error if this bit is set to '1' and SR-IOV Physical functions do not exist.
...	

...

Figure 80: Controller Health Data Structure (CHDS)

Bytes	Description																
...																	
08	Critical Warning (CWARN): This field indicates critical warnings for the state of the Controller. The value of this field corresponds to the value in the Controller's SMART / Health Information Log.																
	<table><tr><th>Bits</th><th>Description</th></tr><tr><td>7:65</td><td>Reserved</td></tr><tr><td>5</td><td>Persistent Memory Region Error (PMRE): This bit is set to '1' when the Persistent Memory Region has become read-only or unreliable.</td></tr><tr><td>4</td><td>Volatile Memory Backup Failed (VMBF): This bit is set to '1' when the volatile memory backup device has failed.</td></tr><tr><td>3</td><td>Read Only (RO): This bit is set to '1' when the media has been placed in read only mode.</td></tr><tr><td>2</td><td>Reliability Degraded (RD): This bit is set to '1' when NVM Subsystem reliability has been degraded due to significant media related errors or an internal error.</td></tr><tr><td>1</td><td>Temperature Above or Under Threshold (TAUT): This bit is set to '1' when a temperature is above an over temperature threshold or below an under-temperature threshold.</td></tr><tr><td>0</td><td>Spare Threshold (ST): This bit is set to '1' when the available spare has fallen below the available spare threshold.</td></tr></table>	Bits	Description	7:65	Reserved	5	Persistent Memory Region Error (PMRE): This bit is set to '1' when the Persistent Memory Region has become read-only or unreliable.	4	Volatile Memory Backup Failed (VMBF): This bit is set to '1' when the volatile memory backup device has failed.	3	Read Only (RO): This bit is set to '1' when the media has been placed in read only mode.	2	Reliability Degraded (RD): This bit is set to '1' when NVM Subsystem reliability has been degraded due to significant media related errors or an internal error.	1	Temperature Above or Under Threshold (TAUT): This bit is set to '1' when a temperature is above an over temperature threshold or below an under-temperature threshold.	0	Spare Threshold (ST): This bit is set to '1' when the available spare has fallen below the available spare threshold.
	Bits	Description															
	7:65	Reserved															
	5	Persistent Memory Region Error (PMRE): This bit is set to '1' when the Persistent Memory Region has become read-only or unreliable.															
	4	Volatile Memory Backup Failed (VMBF): This bit is set to '1' when the volatile memory backup device has failed.															
	3	Read Only (RO): This bit is set to '1' when the media has been placed in read only mode.															
	2	Reliability Degraded (RD): This bit is set to '1' when NVM Subsystem reliability has been degraded due to significant media related errors or an internal error.															
	1	Temperature Above or Under Threshold (TAUT): This bit is set to '1' when a temperature is above an over temperature threshold or below an under-temperature threshold.															
0	Spare Threshold (ST): This bit is set to '1' when the available spare has fallen below the available spare threshold.																
Reserved																	
15:09																	

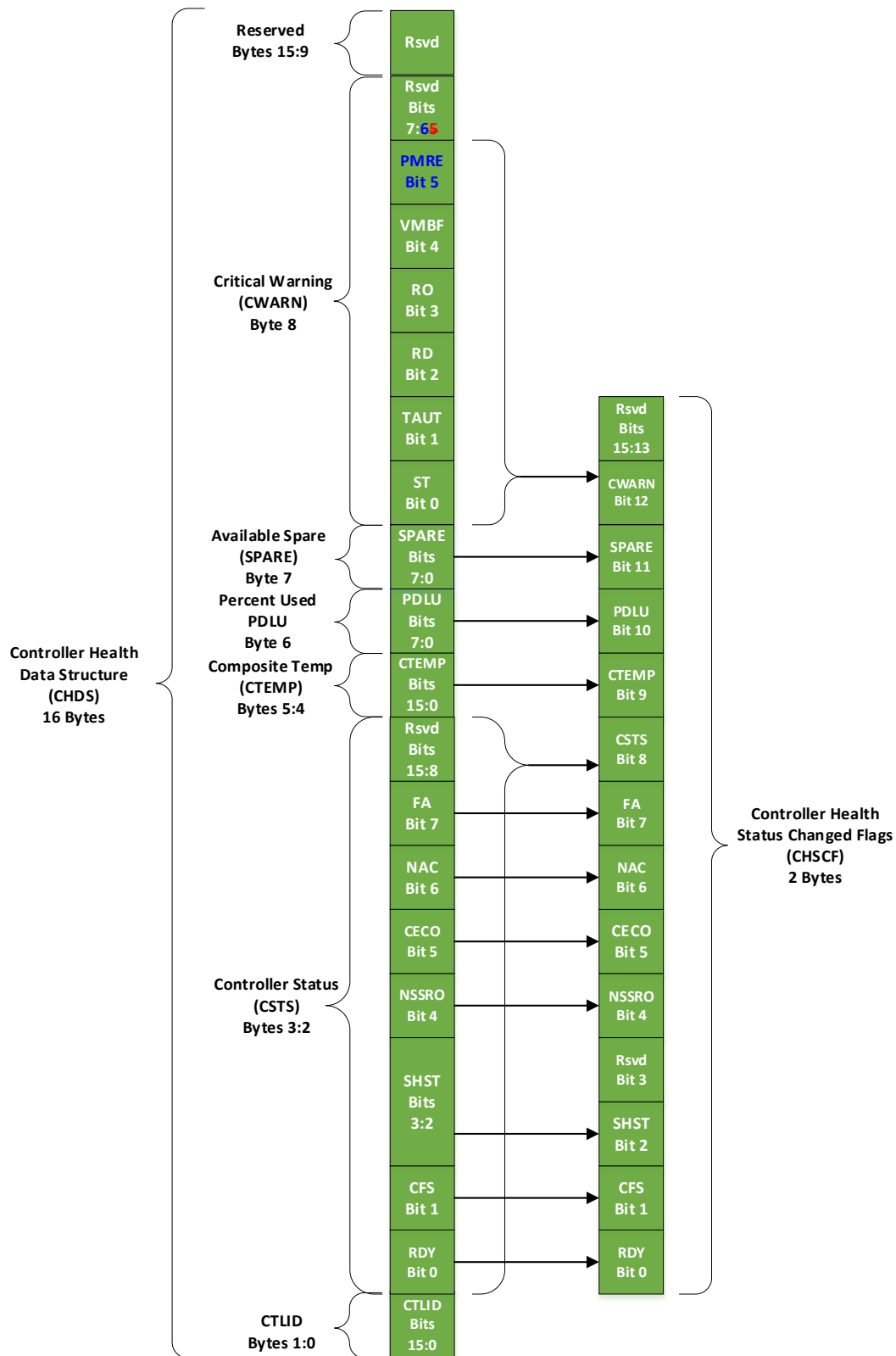
Modify section 5.3.2 as follows:

5.3.2 Filtering by Controller Health Status Changed Flags

The Controller Health Data Structures that are returned by Controller Health Status Poll may also be filtered by the Controller Health Status Changed Flags. Filtering of changes by Controller Health Status Changed Flags is controlled by some of the bits in NVMe Management Dword 1. When one or more of these bits are set to '1' and any of the corresponding bit(s) in the Controller Health Status Changed Flags for the Controller are also set to '1' (refer to Figure 78 for Controller Health Status Changed Flags associated with each bit in NVMe Management Dword 1), then the entire Controller Health Data Structure ~~(including any filtered fields)~~ for that Controller is returned in the Response Data field; else, the Controller Health Data Structure for that Controller is excluded from the Response Data field. The contents returned in the Controller Health Data Structure for filtered fields are undefined.

...

Figure 82: Controller Health Data Structure to Controller Health Status Changed Flags Mapping



Modify several portions of section 5.7 as follows:

5.7 Read NVMe-MI Data Structure

...

Figure 92: Read NVMe-MI Data Structure – NVMe Management Dword 1

Bits	Description
31:08	Reserved
07:00	I/O Command Set Identifier (IOCSI): If the DTYP field value corresponds to Optionally Supported Command List or the Management Endpoint Buffer Command Support List, then this field specifies the I/O Command Set used to select the optional I/O Command Set Specific Admin commands. For more information about I/O Command Sets refer to the NVMe Express Base Specification. For all other values of the DTYP field, this field is reserved. The I/O Command Set specified by this field is not required to be enabled (refer to the NVMe Express Base Specification).

...

Figure 94: NVM Subsystem Information Data Structure

Bytes	Description
00	Number of Ports (NUMP): This field indicates specifies the maximum number of ports of any type supported by the NVM Subsystem. This is a 0's based value. The value of FFh is not supported because a port identifier of 256 is not able to be reported (refer to section 5.1.1).
01	NVMe-MI Major Version Number (MJR): This field shall be set to 1h to indicate the major version number of this specification.
02	NVMe-MI Minor Version Number (MNR): This field shall be set to 2h to indicate the minor version number of this specification.
31:03	Reserved

...

Figure 96: PCIe Port Specific Data

Byte	Description																
08	PCIe Maximum Payload Size: This field indicates the Max Payload Size setting for the specified PCIe port. If the link is not active, this field should be cleared to 0h. <table><tr><th>Value</th><th>Definition</th></tr><tr><td>0h</td><td>128 bytes</td></tr><tr><td>1h</td><td>256 bytes</td></tr><tr><td>2h</td><td>512 bytes</td></tr><tr><td>3h</td><td>1 KiB</td></tr><tr><td>4h</td><td>2 KiB</td></tr><tr><td>5h</td><td>4 KiB</td></tr><tr><td>6h to FFh</td><td>Reserved</td></tr></table> The value reported in this field by ARI Devices and Non-ARI Multi-Function Devices (refer to the PCI Express Base Specification) whose Max Payload Size settings are identical across all Functions is the setting in Function 0. The value reported in this field by non-ARI Multi-Function Devices whose Max Payload Size settings are not identical across all Functions is implementation specific.	Value	Definition	0h	128 bytes	1h	256 bytes	2h	512 bytes	3h	1 KiB	4h	2 KiB	5h	4 KiB	6h to FFh	Reserved
Value	Definition																
0h	128 bytes																
1h	256 bytes																
2h	512 bytes																
3h	1 KiB																
4h	2 KiB																
5h	4 KiB																
6h to FFh	Reserved																
...	...																

...

The Optionally Supported Command List data structure contains a list of optional commands that a Responder supports. The I/O Command Set Identifier (IOCSI) field in NVMe Management Dword 1 selects the I/O Command Set for the I/O Command Set Specific Admin commands that are returned in the Optionally Supported Command List data structure. The Optionally Supported Command List

data structure may contain up to 2,047 commands, and shall be minimally sized (i.e., if there is one optionally supported command, the data structure is 4 bytes total).

Modify several portions of section 5.11 as follows:

5.11 Shutdown

...

Upon receipt of a Shutdown command specifying a Normal NVM Subsystem Shutdown, then: for each Controller in the NVM Subsystem:

- if:
 - CSTS.SHST is cleared to 00b on that Controller; and
 - An outstanding Asynchronous Event Request command exists on that Controller (refer to the NVM Express Base Specification),then the Controller shall issue a Normal NVM Subsystem Shutdown event prior to shutting down the Controller (refer to the NVM Express Base Specification);
- a normal shutdown is initiated on the Controller as specified by the NVM Express Base Specification.

...

The Shutdown command completes successfully when all NVMe Controllers in the NVM Subsystem report shutdown process complete (i.e., CSTS.SHST is set to 10b and CSTS.ST is set to '1'). Refer to the NVMe Express Base Specification on the condition when it is safe to power down the NVM Subsystem.

Modify several portions of section 6 as follows:

6 NVMe Express Admin Command Set

The NVMe Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. Figure 115 shows NVMe Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device and an NVMe Enclosure using the out-of-band mechanism. All NVMe Admin Commands are prohibited using the in-band tunneling mechanism. The commands are defined in the NVM Express Base Specification and the I/O Command Set specifications. If an NVMe Admin Command is issued in a Request Message that is a prohibited command in Figure 115, the Management Endpoint shall return an Invalid Command Opcode Parameter Error Response with PEL field indicating the NVMe opcode. Future revisions of this specification may add additional commands to Figure 115. The NVMe Express Admin Command Set is supported only applicable in the out-of-band mechanism and is prohibited in the in-band tunneling mechanism.

<<Editor, Re-order Figure 115 in Opcode order>>

Figure 115: List of NVMe Admin Commands Supported using the Out-of-Band Mechanism

Command	Opcode	NVMe Storage Device O/M/P ¹	NVMe Enclosure O/M/P ¹	Reference Specification
Abort	00h	P	P	NVMe Express Base Specification
Asynchronous Event Request	0Ch	P	P	NVMe Express Base Specification
Capacity Management	20h	O	P	NVMe Express Base Specification
Create I/O Completion Queue	05h	P	P	NVMe Express Base Specification
Create I/O Submission Queue	01h	P	P	NVMe Express Base Specification
Delete I/O Completion Queue	04h	P	P	NVMe Express Base Specification
Delete I/O Submission Queue	00h	P	P	NVMe Express Base Specification
Device Self-test	14h	O	O	NVMe Express Base Specification
Directive Receive	1Ah	P	P	NVMe Express Base Specification
Directive Send	19h	P	P	NVMe Express Base Specification
Doorbell Buffer Config	7Ch	P	P	NVMe Express Base Specification
Firmware Commit	10h	O	O	NVMe Express Base Specification
Firmware Image Download	11h	O	O	NVMe Express Base Specification
Format NVM	80h	O	P	NVMe Express Base Specification
Get Features	0Ah	M	O	NVMe Express Base Specification
Get LBA Status	86h	O	P	NVM Command Set Specification
Get Log Page ²	02h	M	O	NVMe Express Base Specification
Identify	06h	M	O	NVMe Express Base Specification
Keep Alive	18h	P	P	NVMe Express Base Specification
Lockdown	24h	O	O	NVMe Express Base Specification
Namespace Management	0Dh	O	P	NVMe Express Base Specification
Namespace Attachment	15h	O	P	NVMe Express Base Specification
NVMe-MI Receive	1Dh	P	P	NVMe Express Base Specification
NVMe-MI Send	1Eh	P	P	NVMe Express Base Specification
Sanitize	84h	O	O	NVMe Express Base Specification
Security Send	81h	O	P	NVMe Express Base Specification
Security Receive	82h	O	P	NVMe Express Base Specification
Set Features	09h	O	O	NVMe Express Base Specification
Vendor Specific	C0h to FFh	O	O	NVMe Express Base Specification
Virtualization Management	1Ch	O	O	NVMe Express Base Specification
Fabrics Commands	7Fh	P	P	NVMe Express Base Specification

NOTES:

1. O/M/P definition: O = Optional, M = Mandatory, P = Prohibited from being supported. An NVMe Enclosure that is also an NVMe Storage Device (i.e., implements Namespaces) shall implement mandatory commands required by either an NVMe Storage Device or an NVMe Enclosure and may implement optional commands allowed by either an NVMe Storage Device or an NVMe Enclosure. Mandatory commands shall be supported using the out-of-band mechanism if the NVMe Controller specified by the Controller ID field supports the command in-band.
2. If the Retain Asynchronous Event bit is cleared to '0', then the status associated with the NVMe Admin Command shall be Invalid Field in Command (i.e., the NVMe Admin Command is aborted). For implementations compliant to version 1.1 or earlier of this specification, the Retain Asynchronous Event bit in the Get Log Page command (refer to the NVM Express Base Specification) may or may not be ignored by the Controller. Refer to section 6.2.

NVMe Admin Commands over the out-of-band mechanism may interfere with host software. A Management Controller should coordinate with the host or issue only NVMe Admin Commands that do not interfere with host software or ~~in-band~~ ~~in-band~~ NVMe commands (e.g., Identify). Coordination between a Management Controller and host is outside the scope of this specification.

...

Figure 119: NVMe Admin Command Response Description

Bytes	Description
03:00	NVMe-MI Message Header: Refer to section 3.1.
04	Status: This field indicates the status of the NVMe Admin Command. Refer to section 4.1.2.
07:05	Reserved
11:08	Completion Queue Entry Dword 0 (CQEDW0): Completion Queue Entry Dword 0 as defined in the NVM Express Base Specification.
15:12	Completion Queue Entry Dword 1 (CQEDW1): Completion Queue Entry Dword 1 as defined in the NVM Express Base Specification.
19:16	Completion Queue Entry Dword 3 (CQEDW3): Completion Queue Entry Dword 3 as defined in the NVM Express Base Specification. The Command ID field shall be cleared to 0h.
N-1:20	NVMe Response Data (Optional)
N+3:N	Message Integrity Check: Refer to section 3.1.

Modify figure 122 in section 6.3 to remove redundant (same) information as follows:

6.3 Get Log Page

Figure 122 defines the log pages that are mandatory, optional, and prohibited for SMBus/I2C and PCIe VDM Management Endpoint on NVMe Storage Devices and NVMe Enclosures. [The set of optional log pages supported on each Management Endpoint are allowed to differ \(refer to the NVM Express Base Specification\).](#)

Figure 122: Management Endpoint - Log Page Support

Log Page Name ³	Log Identifier	SMBus/I2C Log Page Support Requirements ¹		PCIe VDM Log Page Support Requirements ¹	
		NVMe Storage Device	NVMe Enclosure	NVMe Storage Device	NVMe Enclosure
Supported Log Pages	00h	M ²	M ²	M ²	M ²
Error Information	01h	M	M	M	M
SMART / Health Information (Controller scope)	02h	M	O	M	O
SMART / Health Information (NVM Subsystem scope)		O	O	O	O
Firmware Slot Information	03h	M	O	M	O
Changed Namespace List	04h	O	O	O	O
Commands Supported and Effects	05h	O	O	O	O
Device Self-test	06h	O	O	O	O
Telemetry Host-Initiated	07h	O	O	O	O
Telemetry Controller-Initiated	08h	O	O	O	O
Endurance Group Information	09h	O	O	O	O
Predictable Latency Per NVM Set	0Ah	O	O	O	O
Predictable Latency Event Aggregate	0Bh	O	O	O	O
Asymmetric Namespace Access	0Ch	O	O	O	O
Persistent Event	0Dh	O	O	O	O
LBA Status Information ⁴	0Eh	O	O	O	O
Endurance Group Event Aggregate	0Fh	O	O	O	O
Media Unit Status	10h	O	O	O	O
Supported Capacity Configuration List	11h	O	O	O	O

Figure 122: Management Endpoint - Log Page Support

Log Page Name ³	Log Identifier	SMBus/I2C Log Page Support Requirements ¹		PCIe VDM Log Page Support Requirements ⁴	
		NVMe Storage Device	NVMe Enclosure	NVMe Storage Device	NVMe Enclosure
Feature Identifiers Supported and Effects	12h	M ²	O	M ²	O
NVMe-MI Commands Supported and Effects	13h	O	O	O	O
Command and Feature Lockdown	14h	O	O	O	O
Boot Partition	15h	O	O	O	O
Rotational Media Information	16h	O	O	O	O
Discovery	70h	O	O	O	O
Reservation Notification	80h	O	O	O	O
Sanitize Status	81h	O	O	O	O
Changed Zone List ⁵	BFh	O	O	O	O
Notes: 1. O = Optional, M = Mandatory, P = Prohibited. 2. Optional for versions 1.1 and earlier of this specification. 3. Refer to the NVM Express Base Specification unless another footnote specifies otherwise. 4. Refer to the NVM Command Set Specification. 5. Refer to the Zoned Namespace Command Set Specification.					

Modify the title of section 6.4 to align with the title change defined by TP 4047b as follows:

6.4 Sanitize Operation and Format NVM Command

Modify figure 126 in section 6.5 to retain the support requirements for HOST Metadata features defined by NVMe-MI 1.1 and TP 6009. Also remove redundant column information:

6.5 Set Features and Get Features

...

Figure 126 defines the features that are mandatory, optional, and prohibited for SMBus/I2C and PCIe VDM Management Endpoints on NVMe Storage Devices and NVMe Enclosures. The set of optional features supported on each Management Endpoint are allowed to differ (refer to the NVM Express Base Specification).

Figure 126: Management Endpoint - Feature Support

Feature Name ²	Feature Identifier	SMBus/I2C Feature Support Requirements ¹		PCIe VDM Feature Support Requirements ¹	
		NVMe Storage Device	NVMe Enclosure	NVMe Storage Device	NVMe Enclosure
Arbitration	01h	P	P	P	P
Power Management	02h	O	O	O	O

Figure 126: Management Endpoint - Feature Support

Feature Name ²	Feature Identifier	SMBus/I2C Feature Support Requirements ¹		PCIe VDM Feature Support Requirements ¹	
		NVMe Storage Device	NVMe Enclosure	NVMe Storage Device	NVMe Enclosure
LBA Range Type ³	03h	P	P	P	P
Temperature Threshold	04h	O	O	O	O
Error Recovery ³	05h	P	P	P	P
Volatile Write Cache	06h	P	P	P	P
Number of Queues	07h	P	P	P	P
Interrupt Coalescing	08h	P	P	P	P
Interrupt Vector Configuration	09h	P	P	P	P
Write Atomicity Normal ³	0Ah	P	P	P	P
Asynchronous Event Configuration	0Bh	P	P	P	P
Autonomous Power State Transition	0Ch	O	O	O	O
Host Memory Buffer	0Dh	P	P	P	P
Timestamp	0Eh	O	O	O	O
Keep Alive Timer	0Fh	P	P	P	P
Host Controlled Thermal Management	10h	O	O	O	O
Non-Operational Power State Config	11h	O	O	O	O
Read Recovery Level Config	12h	P	P	P	P
Predictable Latency Mode Config	13h	P	P	P	P
Predictable Latency Mode Window	14h	P	P	P	P
LBA Status Information Attributes ³	15h	P	P	P	P
Host Behavior Support	16h	P	P	P	P
Sanitize Config	17h	O	O	O	O
Endurance Group Event Configuration	18h	P	P	P	P
I/O Command Set Profile	19h	O	P	O	P
Spinup Control	1Ah	O	O	O	O
Key Value Configuration ⁴	20h	O	O	O	O
Enhanced Controller Metadata	7Dh	OM	OM	O	O
Controller Metadata	7Eh	OM	OM	O	O
Namespace Metadata	7Fh	OM	O	O	O
Software Progress Marker	80h	P	P	P	P
Host Identifier	81h	P	P	P	P
Reservation Notification Mask	82h	P	P	P	P
Reservation Persistence	83h	P	P	P	P
Namespace Write Protection Config	84h	P	P	P	P
Notes: 1. O = Optional, M = Mandatory, P = Prohibited for Set Features/Optional for Get Features. 2. Refer to the NVM Express Base Specification unless another footnote specifies otherwise. 3. Refer to the NVM Command Set Specification. 4. Refer to the Key Value Command Set Specification.					

Modify section 8.2.3 as follows:

8 Management Architecture

...

8.2 Vital Product Data

...

8.2.3 NVMe MultiRecord Area

This MultiRecord is used to describe the form factor, power requirements, and capacity of NVMe Storage Devices with a single NVM Subsystem. Implementations compliant to version 1.1 and later of this specification should implement the Topology MultiRecord (refer to section 8.2.5). For backwards compatibility, the NVMe MultiRecord and the NVMe PCIe Port MultiRecord (refer to section 8.2.4) should both be included in the VPD in addition to the Topology MultiRecord unless:

a) the NVMe Storage Device FRU has:

1. Expansion Connectors; ~~or~~₇
2. more than one NVM Subsystem;₇

or

b) if including both this MultiRecord and the NVMe PCIe Port MultiRecord would extend the size of the VPD beyond 256 bytes.

If either the NVMe MultiRecord or NVMe PCIe Port MultiRecord ~~is are~~ not included, then neither MultiRecord should be included.

Figure 152: NVMe PCIe Port MultiRecord Area

Bytes	Factory Default	Description	
00	0Ch	NVMe PCIe Port Record Type ID	
01	02h or 82h	Record Format:	
		Bits	Definitionhas:
		7	Set to '1' if last record in list.
		6:0	Record format version = shall be set to 2h.
...			

Modify section 8.2.4 as follows:

8.2.4 NVMe PCIe Port MultiRecord Area

This MultiRecord is used to describe the PCIe connectivity for NVMe Storage Devices with a single NVM Subsystem. Implementations compliant to version 1.1 and later of this specification should implement the Topology MultiRecord (refer to section 9.2.5). For backwards compatibility, the NVMe PCIe Port MultiRecord and the NVMe MultiRecord (refer to section 8.2.3) should both be included in the VPD in addition to the Topology MultiRecord unless:

a) the NVMe Storage Device FRU has:

1. Expansion Connectors; ~~or~~₇
2. more than one NVM Subsystem;₇

or

- b) if including both this MultiRecord and the NVMe MultiRecord would extend the size of the VPD beyond 256 bytes.

If either the NVMe MultiRecord or NVMe PCIe Port MultiRecord ~~is are~~ not included then neither MultiRecord should be included.

Figure 153: NVMe PCIe Port MultiRecord Area

Bytes	Factory Default	Description	
00	0Ch	NVMe PCIe Port Record Type ID	
01	02h or 82h	Record Format:	
		Bits	Definition
		7	Set to '1' if last record in list.
		6:0	Record format version - shall be set to 2h.
02	08h or 0Bh	Record Length (RLEN): This field indicates the length of the MultiRecord Area in bytes without including the first 5 bytes that are common to all MultiRecords.	
...			

Modify section 8.2.5.1 as follows:

8.2.5 Topology MultiRecord Area

...

8.2.5.1 Extended Element Descriptor

...

Figure 158: Extended Element Descriptor

Bytes	Factory Default	Description
00	01h	Type: This field indicates the type of the Element Descriptor. This field shall be set to tThe Extended Element Descriptor Type is (i.e., 1h). Refer to Figure 157.
01	00h	Revision: This field indicates the revision of the Element Descriptor. The Extended Element Descriptor Revision. This field shall be cleared to is 0h for this specification.
02	Impl Spec	Length: This field indicates the length of the Extended Element Descriptor in bytes.
Length - 1:03	Impl Spec	Extended Content: This field extends the content of the Element Descriptor at the immediately preceding index.

Modify section 8.2.5.2 as follows:

8.2.5.2 Upstream Connector Element Descriptor

The Upstream Connector Element Descriptor is shown in Figure 159 and is used to describe an Upstream Connector (i.e., a connector through which a Requester communicates with the NVMe Storage Device). Upstream Element Descriptors are always a parent and never a child.

Figure 159: Upstream Connector Element Descriptor

Bytes	Factory Default	Description
00	02h	Type: This field indicates the type of the Element Descriptor. This field shall be set to tThe Upstream Connector Element Descriptor Type is (i.e., 2h). Refer to Figure 157.

Figure 159: Upstream Connector Element Descriptor

Bytes	Factory Default	Description
01	00h	Revision: This field indicates the revision of the Element Descriptor. The Upstream Connector Element Descriptor Revision. This field shall be cleared to is 0h for this specification.
...		

...

Figure 161: SMBus/I2C Upstream Port Descriptor

Bytes	Factory Default	Description
00	00h	Type: This field indicates the type of the Port Descriptor. This field shall be cleared to The SMBus/I2C Upstream Port Descriptor Type is 0h.
01	Impl Spec	Length: This field indicates the length of the SMBus/I2C Upstream Port Descriptor in bytes.
...

...

Figure 162: PCIe Upstream Port Descriptor

Bytes	Factory Default	Description
00	01h	Type: This field indicates the type of Upstream Port Descriptor. This field shall be set to The PCIe Upstream Port Descriptor Type is 1h.
...

Modify section 8.2.5.3 as follows:

8.2.5.3 Expansion Connector Element Descriptor

...

Figure 163: Expansion Connector Element Descriptor

Bytes	Factory Default	Description
00	03h	Type: This field indicates the type of the Element Descriptor. This field shall be set to t The Expansion Connector Element Descriptor Type is (i.e., 3h). Refer to Figure 157.
01	00h	Revision: This field indicates the revision of the Element Descriptor. The Expansion Connector Element Descriptor Revision. This field shall be cleared to is 0h for this specification.
...

...

Figure 164: Expansion Connector PCIe Port Descriptor

Bytes	Factory Default	Description
00	00h	Type: This field indicates the type of Expansion Connector Port Descriptor. This field shall be cleared to The Expansion Connector PCIe Port Descriptor Type is 0h.
...		

Modify section 8.2.5.4 as follows:

8.2.5.4 Label Element Descriptor

...

Figure 165: Label Element Descriptor

Bytes	Factory Default	Description
00	04h	Type: This field indicates the type of the Element Descriptor. This field shall be set to 4. The Label Element Descriptor Type is (i.e., 4h). Refer to Figure 157.
01	00h	Revision: This field indicates the revision of the Element Descriptor. The Label Element Descriptor Revision. This field shall be cleared to 0h for this specification.
02	Impl Spec	Length: This field indicates the length of the Label Element Descriptor in bytes including the null termination.
Length - 1:03	Impl Spec	Label String: This field contains a null-terminated UTF-8 string used to identify the parent Element Descriptor.

Modify section 8.2.5.5 as follows:

8.2.5.5 SMBus/I2C Mux Element Descriptor

...

Figure 166: SMBus/I2C Mux Element Descriptor

Bytes	Factory Default	Description
00	05h	Type: This field indicates the type of the Element Descriptor. This field shall be set to 5. The SMBus/I2C Mux Element Descriptor Type is (i.e., 5h). Refer to Figure 157.
01	00h	Revision: This field indicates the revision of the Element Descriptor. The SMBus/I2C Mux Element Descriptor Revision. This field shall be cleared to 0h for this specification.
02	Impl Spec	Length: This field indicates the length of the SMBus/I2C Mux Element Descriptor in bytes.
...

...

Figure 168: SMBus/I2C Mux Channel Descriptor

Bytes	Factory Default	Description
00	00h	Type: This field indicates the type of the Descriptor. The SMBus/I2C Mux Channel Descriptor. This field shall be cleared to 0h.
...

Modify section 8.2.5.6 as follows:

8.2.5.6 PCIe Switch Element Descriptor

...

Figure 169: PCIe Switch Element Descriptor

Bytes	Factory Default	Description
00	06h	Type: This field indicates the type of the Element Descriptor. This field shall be set to 6h. The PCIe Switch Element Descriptor Type is (i.e., 6h). Refer to Figure 157.
01	Impl Spec	Revision: This field indicates the revision of the Element Descriptor. The PCIe Switch Element Descriptor Revision. This field shall be cleared to 0h for this specification.
...

...

Figure 170: PCIe Switch Port Descriptor

Bytes	Factory Default	Description
00	00h	Type: This field indicates the type of Port Descriptor. The PCIe Switch Port Descriptor. This field shall be cleared to 0h.
...

Modify section 8.2.5.7 as follows:

8.2.5.7 NVM Subsystem Element Descriptor

...

Figure 171: NVM Subsystem Element Descriptor

Bytes	Factory Default	Description					
00	07h	Type: This field indicates the type of the Element Descriptor. This field shall be set to 7h. The NVM Subsystem Element Descriptor Type is (i.e., 7h). Refer to Figure 157.					
01	00h	Revision: This field indicates the revision of the Element Descriptor. The NVM Subsystem Element Descriptor Revision. This field shall be cleared to is 0h for this specification.					
02	Impl Spec	Length: This field indicates the length of the NVM Subsystem Element Descriptor in bytes.					
03	3Ah or 3Bh	SMBus/I2C Address Info: If the NVM Subsystem supports an MCTP over SMBus/I2C port, then this field indicates the SMBus/I2C address for MCTP over SMBus/I2C port and whether or not SMBus ARP is supported; otherwise, this field shall be cleared to has a value of 0h.					
		Bits	Description	7:1	SMBus/I2C Address: This field contains the 7-bit SMBus/I2C address. Refer to Figure 16 for requirements.	0	ARP Capable: This bit is set to '1' if SMBus ARP is supported, else it is cleared to '0'. Refer to Figure 16 for requirements.
		Bits	Description				
7:1	SMBus/I2C Address: This field contains the 7-bit SMBus/I2C address. Refer to Figure 16 for requirements.						
0	ARP Capable: This bit is set to '1' if SMBus ARP is supported, else it is cleared to '0'. Refer to Figure 16 for requirements.						

Figure 171: NVM Subsystem Element Descriptor

Bytes	Factory Default	Description		
04	Impl Spec	SMBus/I2C Capabilities: If the NVM Subsystem supports an SMBus/I2C port then this field indicates the SMBus/I2C capabilities; otherwise, this field shall be cleared to has a value of 0h.		
		Bits	Description	
		7	Reset: This bit is set to '1' if all of the SMBus/I2C reset mechanisms are supported as defined by the associated form factor specification. This bit is cleared to '0' if the form factor does not define SMBus Reset or the NVMe Storage Device does not support all of the SMBus/I2C reset mechanisms defined by the specification for the Form Factor in the Host Connector Element Descriptor.	
		6:2	Reserved	
		1:0	Maximum Speed: This field is set to the highest supported SMBus/I2C clock speed.	
			Value	Description
0	100 kHz			
1	400 kHz			
2	1 MHz			
	3	Reserved		
...		

...

Figure 172: NVM Subsystem Port Descriptor

Bytes	Factory Default	Description
00	00h	Type: This field indicates the type of an NVM Subsystem Port Descriptor. This field shall be cleared to The NVM Subsystem Port Descriptor Type is 0h.
...

Modify section 8.2.5.8 as follows:

8.2.5.8 FRU Information Device Element Descriptor

...

Figure 173: FRU Information Device Element Descriptor

Byte Offset	Factory Default	Description
00	08h	Type: This field indicates the type of the Element Descriptor. This field shall be set to The FRU Information Device Element Descriptor Type is (i.e., 8h). Refer to Figure 157.
01	00h	Revision: This field indicates the revision of the Element Descriptor. The FRU Information Device Element Descriptor Revision. This field shall be cleared to is 0h for this specification.
...

Modify section 8.3.1 as follows:

8.3 Reset

...

8.3.1 NVM Subsystem Reset

An NVM Subsystem Reset is initiated under the conditions outlined in the NVM Express Base Specification (e.g., when main power is applied to the NVM Subsystem). In addition to these conditions, if NVM Subsystem Reset is supported, then it may be initiated by:

- processing a Reset command; or
- processing a PCIe Memory Write command (refer to section 7.6) to the NSSR property (refer to the NVM Express Base Specification) that specifies the value 4E564D65h ("NVMe").

Modify section Appendix A as follows

Appendix A Technical Note: NVM Express Basic Management Command

...

Figure 175: Subsystem Management Data Structure

Command Code	Offset (byte)	Description
0	00	Length of Status: Indicates number of additional bytes to read before encountering PEC. This value should always be 6 (06h) in implementations of this version of the spec.
	01	Status Flags (SFLGS): This field indicates the status of the NVM Subsystem. SMBus Arbitration – Bit 7 is set to '1' after an SMBus block read is completed all the way to the stop bit without bus contention and cleared to '0' if an SMBus Send Byte FFh is received on this SMBus address. Drive Not Ready – Bit 6 is set to '1' when the NVMe Subsystem is not capable of processing NVMe management commands, and the rest of the transmission may be invalid. If cleared to '0', then the NVM Subsystem is fully powered and ready to respond to management commands. This logic level intentionally identifies and prioritizes powered up and ready drives over their powered off neighbors on the same SMBus channel. Drive Functional – Bit 5 is set to '1' to indicate an NVM Subsystem is functional. If cleared to '0', then there is an unrecoverable failure in the NVM Subsystem and the rest of the transmission may be invalid. Note that this bit may default to '0' after reset and transition to '1' after the NVM Subsystem has completed initialization and this case should not be considered an error. Reset Not Required - Bit 4 is set to '1' to indicate the NVM Subsystem does not require a reset to resume normal operation. If cleared to '0', then the NVM Subsystem has experienced an error that prevents continued normal operation. A Controller Level Reset is required to resume normal operation. Port 0 PCIe Link Active - Bit 3 is set to '1' to indicate the first port's PCIe link is up (i.e., the Data Link Control and Management State Machine is in the DL_Active state). If cleared to '0', then the PCIe link is down. Port 1 PCIe Link Active - Bit 2 is set to '1' to indicate the second port's PCIe link is up. If cleared to '0', then the second port's PCIe link is down or not present. Bits 1:0 shall be set to 11b.

Figure 175: Subsystem Management Data Structure

Command Code	Offset (byte)	Description
	02	<p>SMART Warnings: This field shall contain the Critical Warning field (byte 0) of the NVMe SMART / Health Information log. Each bit in this field shall be inverted from the NVMe definition (i.e., the management interface shall indicate a '0' value while the corresponding bit is '1' in the log page). Refer to the NVMe Express Base Specification for bit definitions.</p> <p>If there are multiple Controllers in the NVM Subsystem, the Management Endpoint shall combine the Critical Warning field from every Controller such that a bit in this field is:</p> <ul style="list-style-type: none"> • Cleared to '0' if any Controller in the NVMe Subsystem indicates a critical warning for that corresponding bit. • Set to '1' if all Controllers in the NVM Subsystem do not indicate a critical warning for the corresponding bit.
	...	
...		