



#### **LEGAL NOTICE:**

© **Copyright 2007 to 2020 NVM Express™, Inc. ALL RIGHTS RESERVED.**

This erratum to the NVM Express Management Interface revision 1.1 specification is proprietary to the NVM Express, Inc. (also referred to as "Company") and/or its successors and assigns.

**NOTICE TO USERS WHO ARE NVM EXPRESS, INC. MEMBERS:** Members of NVM Express, Inc. have the right to use and implement this erratum to the NVM Express Management Interface revision 1.1 specification subject, however, to the Member's continued compliance with the Company's Intellectual Property Policy and Bylaws and the Member's Participation Agreement.

**NOTICE TO NON-MEMBERS OF NVM EXPRESS, INC.:** If you are not a Member of NVM Express, Inc. and you have obtained a copy of this document, you only have a right to review this document or make reference to or cite this document. Any such references or citations to this document must acknowledge NVM Express, Inc. copyright ownership of this document. The proper copyright citation or reference is as follows: "© **2007 to 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.

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

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

## NVM Express™ Technical Errata

<b>Errata ID</b>	<b>002</b>
<b>Revision Date</b>	<b>2020-09-16</b>
<b>Affected Spec Ver.</b>	<b>NVM Express Management Interface 1.1a</b>
<b>Corrected Spec Ver.</b>	<b>NVM Express Management Interface 1.1a+</b>

### Errata Author(s)

<b>Name</b>	<b>Company</b>
Austin Bolen	Dell EMC
Mike Allison	Intel

### Errata Overview

- Replaced the term “execute” with “process” in relation to command process.
- Clarified the alignment of the Message Integrity Check (MIC) field.
- Added a recommendation that an Invalid Parameter Error Response be transmitted if CSI bit is set to ‘1’ on a Pause Control Primitive.
- Corrected the PCIe VDM Maximum MCTP Transmission Unit Size.
- Replaced “lower/upper” with “least/most significant”
- Removed conflicting statement that said if a PCIe port is in PCIe Conventional Reset, Management Endpoints associated with other PCIe ports not in PCIe Conventional Reset or an SMBus/I2C not in SMBus/I2C reset may not allow access.
- Clarified the requirement of a Management Endpoint reset not affecting other Management Endpoints.
- Applied a diagram from NVMe-MI 1.0 ECN 002 that was not integrated.

## Revision History

Revision Date	Author	Change Description
2020-02-23	Austin Bolen	<ul style="list-style-type: none"><li>Initial draft.</li><li>Corrected the PCIe VDM Maximum MCTP Transmission Unit Size.</li></ul>
2020-02-24	Austin Bolen	<ul style="list-style-type: none"><li>Updated errata regarding PCIe VDM Maximum MCTP Transmission Unit Size.</li><li>Added clarifications on how DOFST, DLEN, and CFLAGs bits 0 and 1 work together.</li></ul>
2020-02-28	Austin Bolen	<ul style="list-style-type: none"><li>Removed conflicting statement that said if a PCIe port is in PCIe Conventional Reset, Management Endpoints associated with other PCIe ports not in PCIe Conventional Reset or an SMBus/I2C not in SMBus/I2C reset may not allow access.</li><li>Updated the section on how DOFST, DLEN, and CFLAGs bits 0 and 1 work together based on feedback from Karthik Ranganathan.</li></ul>
2020-04-13	Mike Allison	<ul style="list-style-type: none"><li>Updated the backward incompatible section</li><li>Removed “upper” and “lower”</li><li>Reformatted to get heading pages updated with section numbers</li></ul>
2020-04-20	Austin Bolen	<ul style="list-style-type: none"><li>Replaced the work “executing” with “processing”</li></ul>
2020-05-11	Mike Allison	<ul style="list-style-type: none"><li>Added clarity on the MIC alignment</li><li>Added recommendation on CSI set to ‘1’ on a Pause Primitive</li><li>Updated appendix C to include missing case from NVM-MI 1.0 ECN 002.</li></ul>
2020-06-15	Mike Allison	<ul style="list-style-type: none"><li>Updated the list of changes. Added comments to the DOSFT to be discussed.</li></ul>
2020-06-22	Mike Allison	<ul style="list-style-type: none"><li>Re-worded the alignment text on the MIC field. Removed section 6 as that text needs to be a TPAR.</li></ul>
2020-06-29	Mike Allison	<ul style="list-style-type: none"><li>Renamed the filename for the member review.</li></ul>
2020-08-17	Jarryd Allison	<ul style="list-style-type: none"><li>Integrated into NVMe-MI Base Specification version 1.1a</li></ul>
2020-08-24	Mike Allison	<ul style="list-style-type: none"><li>Accepted all changes and removed all comments for ratification.</li></ul>
2020-09-16	Mike Allison	<ul style="list-style-type: none"><li>Fixed fonts and spelling of “ENC” to “ECN”.</li></ul>

## Incompatible Changes

- The Maximum MCTP Transmission Unit Size field in the *Port Information Data Structure* incorrectly subtracted the header in the definition of the maximum size. The subtraction is removed.

## Markup Conventions:

Black:	Unchanged (however, hot links are removed)
<del>Red Strikethrough:</del>	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 a portion of 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~~ ~~executes~~ Controller operations on behalf of any Controller (in-band tunneling mechanism) or Management Endpoint (out-of-band mechanism) in the NVM Subsystem. Controllers or Management Endpoints may route commands to any NVMe Controller in the NVM Subsystem. A Controller Management Interface logically ~~processes~~ ~~executes~~ one operation at a time. A Controller Management Interface is not precluded from ~~processing~~ ~~executing~~ two or more operations in parallel; however, there shall always be an equivalent pattern of sequential operations with the same results.

*Modify a portion of section 1.8.6 as follows:*

### 1.8.6 NVMe Controller Management Interface (Controller Management Interface)

An interface associated with each NVMe Controller in the NVM Subsystem that is responsible for ~~processing~~ ~~executing~~ management operations on behalf of a Management Endpoint.

*Modify a portion Figure 18 in section 3.1.1 as follows:*

### 3.1.1 Message Fields

...

**Figure 1: NVMe-MI Message Fields**

Byte	Description
...	
N+3:N	<b>Message Integrity Check (MIC):</b> If the Integrity Check (IC) bit is set to '1', then this field contains a CRC computed over the contents of the NVMe-MI Message. Refer to section 3.1.1.1. If the IC bit is cleared to '0', then this field is not included in the NVMe-MI Message. <a href="#">This field is byte aligned.</a>

*Modify a portion Figure 26 in section 4.1.2 as follows:*

### 4.1.2 Response Messages

...

**Figure 2: Response Message Status Values**

Value	Description	Error Response Format Section
...		
02h	<b>Internal Error:</b> The Request Message could not be <del>processed</del> <del>executed</del> due to a vendor specific internal error.	4.1.2.1

*Modify Figure 32 in section 4.2.1 as follows:*

#### 4.2.1 Control Primitives

...

**Figure 3: Control Primitive Fields**

Byte	Description
03:00	<b>NVMe-MI Message Header (NMH):</b> Refer to section 3.1.
04	<b>Control Primitive Opcode (CPO):</b> This field specifies the opcode of the Control Primitive to be processed <del>executed</del> . Refer to Figure 33.

*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 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.

*Modify Figure 38 in section 4.2.1.4 as follows:*

##### 4.2.1.4 Get State

...

**Figure 4: Get State Control Primitive Success Response Fields**

Byte	Description		
07:06	<b>Control Primitive Specific Response (CPSR):</b> This field is used to return Control Primitive specific status.		
	Bits	CS Specific <sup>1</sup>	Description
	15	Yes	<b>Pause Flag (PFLG):</b> This bit indicates whether or not the Command Slot is paused. This bit set to '1' indicates the Command Slot is paused. This bit cleared to '0' indicates the Command Slot is not paused.  While the Pause Flag is set, the Management Endpoint disables the timeout waiting for packet timer, as defined in the MCTP Base Specification, for the Command Slot and does not transmit responses to Command Messages.
	14	No	<b>NVM Subsystem Reset Occurred (NSSRO):</b> This bit indicates when an NVM Subsystem Reset occurs while main power is applied. This bit is set to '1' if the last occurrence of an NVM Subsystem Reset occurred while main power was applied to the NVM Subsystem. This bit is cleared to '0' following a power cycle and following a Get State primitive with the CESF bit set to '1'.
	13	No	<b>Bad Packet or Other Physical Layer (BPOPL):</b> This bit is set to '1' if a packet sent to the Management Endpoint failed a transport specific packet integrity check since the last time Get State primitive was processed <del>executed</del> with the CESF bit set to '1'.
	12	No	<b>Bad, Unexpected, or Expired Message Tag (BUENT):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was processed <del>executed</del> with the CESF bit set to '1'.

**Figure 4: Get State Control Primitive Success Response Fields**

Byte	Description												
	11	No	<b>Out-of-Sequence Packet Sequence Number (OSPSN):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	10	No	<b>Unexpected Middle or End of Packet (UMEP):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	09	No	<b>Incorrect Transmission Unit (ITU):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	08	No	<b>Unknown Destination ID (UDSTID):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	07	No	<b>Bad Header Version (BHVS):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	06	No	<b>Unsupported Transmission Unit (UTUNT):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	05	No	<b>Timeout Waiting for a Packet (WPTT):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	04	No	<b>Bad Message Integrity Check Error (BMICE):</b> This bit is set to '1' if the Management Endpoint detected an error of this type (refer to the MCTP Base Specification) since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	03	No	<b>Command Message to non-Idle Command Slot (CMNICS):</b> This bit is set to '1' if the Management Endpoint discarded one or more Command Messages due to overlapping Command Messages to a Command Slot since the last time Get State primitive was <b>processed</b> <del>executed</del> with the CESF bit set to '1'.										
	02		Reserved										
01:00	Yes	<b>Slot Command Servicing State (SSTA):</b> This field indicates the current command servicing state of the Command Slot. An implementation may choose to indicate only the Idle and Process states in this field. Refer to Figure 30.	<table><tr><th>Value</th><th>Description</th></tr><tr><td>0h</td><td>Idle</td></tr><tr><td>1h</td><td>Receive</td></tr><tr><td>2h</td><td>Process</td></tr><tr><td>3h</td><td>Transmit</td></tr></table>	Value	Description	0h	Idle	1h	Receive	2h	Process	3h	Transmit
			Value	Description									
			0h	Idle									
			1h	Receive									
			2h	Process									
3h	Transmit												
NOTES:													
1. Command Slot Specific. A 'Yes' in this column indicates the value of the field is independent per Command Slot within a Management Endpoint.													

*Modify a portion of section 4.2.3 as follows:*

#### 4.2.3 Management Endpoint Buffer

...

The Management Endpoint Buffer is considered a cache in the context of sanitize operations performed in an NVM Subsystem. The MCTP Management Endpoint Buffer may contain Response Data associated with a

previously ~~processed~~ ~~executed~~ command that is not allowed during a sanitize operation. When a sanitize operation is initiated, the contents of the Management Endpoint Buffer shall be cleared to 0h. An attempt to access this zeroed data by a Management Endpoint Buffer Read command or any Command Message that uses the Management Endpoint Buffer, then the Management Endpoint responds with a Response Message Status of Management Endpoint Buffer Cleared Due to Sanitize. This Response Message Status is commonly associated with a Management Endpoint Buffer Read command but may be associated with any command that uses the Management Endpoint Buffer as Request Data.

*Modify a portion figure 54 in section 5 as follows:*

## 5 Management Interface Command Set

...

**Figure 5: NVMe-MI Command Request Message Description (NCREQ)**

Byte	Description
03:00	<b>NVMe-MI Message Header (NMH):</b> Refer to section 3.1.
04	<b>Opcode (OPC):</b> This field specifies the opcode of the NVMe-MI Command to be <del>processed</del> <del>executed</del> . Refer to Figure 55.

*Modify a portion of section 5.3 as follows:*

### 5.3 Controller Health Status Poll

...

The Controller Health Data Structure, shown in Figure 77, contains the health status attributes that are tracked for each Controller. When the command is ~~processed without error~~ ~~executed~~, health status is returned for up to 255 Controllers starting at or above the Starting Controller ID (SCTLID). Controllers are returned in ascending order of Controller Identifier starting at offset 0h of the Response Data.

*Modify Figure 91 (Port Information Data Structure) in section 5.7 as follows:*

### 5.7 Read NVMe-MI Data Structure

...

Byte	Description
...	...
03:02	<p><b>Maximum MCTP Transmission Unit Size:</b> The maximum MCTP Transmission Unit size the port is capable of sending and receiving.</p> <p>If the port does not support MCTP, then this field shall be cleared to 0h.</p> <p>If the <del>p</del>Port <del>t</del>Type is PCIe and the port supports MCTP, then this field shall be set to a value between 64 bytes and the PCIe Max Payload Size <del>s</del>Supported (refer to the PCI Express Base Specification) <del>minus 4</del>, inclusive. All PCIe ports within an NVM Subsystem should report the same value in this field.</p> <p>If the <del>p</del>Port <del>t</del>Type is SMBus and the port supports MCTP, then this field shall be set to a value between 64 bytes and 250 bytes, inclusive.</p>
...	...

*Modify portions of section 5.12 as follows:*

## 5.12 VPD Write

...

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

The Requester should not read the contents of the VPD while this command is servicing. Reading the contents of the VPD or **the processing of** ~~executing~~ a VPD Read command while a VPD Write command is **being processed** ~~executing~~ may return incorrect data as a result of the read.

*Modify a figure 118 and portion of section 7 as shown below:*

## 7 PCIe Command Set (Optional)

...

**Figure 6: PCIe Command Request Description**

Byte	Description
03:00	<b>NVMe-MI Message Header (NMH):</b> Refer to section 3.1.
04	<b>Opcode (OPC):</b> This field specifies the opcode of the command to be <b>processed</b> <del>executed</del> . Refer to Figure 119.

...

It is recommended that PCIe Commands provide access to all non-blocked address spaces whenever MCTP access is supported. In some implementations, it may not be possible to access PCIe resources in certain states. A PCIe Command **processed** ~~executed~~ when a Controller is in one of these states may be aborted with the Status field set to PCIe Inaccessible. Refer to section 0.

*Modify a portion of Figure 126 in section 7.3 as shown below:*

### 7.3 PCIe I/O Read

...

**Figure 7: PCIe I/O Read – PCIe Request Dword 0**

Bit	Description
31:19	Reserved

**Figure 7: PCIe I/O Read – PCIe Request Dword 0**

Bit	Description																
18:16	<p><b>Base Address Register (BAR):</b> This field specifies the PCI Base Address Register (BAR) of the I/O space to be read. BARs are located beginning at 10h in PCI Configuration space and the value of this field specifies the starting offset of the associated BAR. For a 64-bit BAR, this field should correspond to the <del>least significant lower</del> 32-bits of the BAR.</p> <table> <tr> <th>Value</th><th>BAR Offset</th></tr> <tr> <td>0h</td><td>10h</td></tr> <tr> <td>1h</td><td>14h</td></tr> <tr> <td>2h</td><td>18h</td></tr> <tr> <td>3h</td><td>1Ch</td></tr> <tr> <td>4h</td><td>20h</td></tr> <tr> <td>5h</td><td>24h</td></tr> <tr> <td>6h to 7h</td><td>Reserved</td></tr> </table>	Value	BAR Offset	0h	10h	1h	14h	2h	18h	3h	1Ch	4h	20h	5h	24h	6h to 7h	Reserved
Value	BAR Offset																
0h	10h																
1h	14h																
2h	18h																
3h	1Ch																
4h	20h																
5h	24h																
6h to 7h	Reserved																
15:00	<b>Length (LENGTH):</b> This field specifies the number of bytes to be read.																

*Modify a portion of Figure 126 in section 7.4 as shown below:*

#### 7.4 PCIe I/O Write

...

**Figure 8: PCIe I/O Write – PCIe Request Dword 0**

Bit	Description																
31:19	Reserved																
18:16	<p><b>Base Address Register (BAR):</b> This field specifies the PCI Base Address Register (BAR) of the I/O space to be written. BARs are located beginning at 10h in PCI Configuration space and the value of this field specifies the starting offset of the associated BAR. For a 64-bit BAR, this field should correspond to the <del>least significant lower</del> 32-bits of the BAR.</p> <table> <tr> <th>Value</th><th>BAR Offset</th></tr> <tr> <td>0h</td><td>10h</td></tr> <tr> <td>1h</td><td>14h</td></tr> <tr> <td>2h</td><td>18h</td></tr> <tr> <td>3h</td><td>1Ch</td></tr> <tr> <td>4h</td><td>20h</td></tr> <tr> <td>5h</td><td>24h</td></tr> <tr> <td>6h to 7h</td><td>Reserved</td></tr> </table>	Value	BAR Offset	0h	10h	1h	14h	2h	18h	3h	1Ch	4h	20h	5h	24h	6h to 7h	Reserved
Value	BAR Offset																
0h	10h																
1h	14h																
2h	18h																
3h	1Ch																
4h	20h																
5h	24h																
6h to 7h	Reserved																
15:00	<b>Length (LENGTH):</b> This field specifies the number of bytes to be written.																

*Modify a portion of Figure 131 and Figure 132 in section 7.5 as shown below:*

#### 7.5 PCIe Memory Read

...

**Figure 9: PCIe Memory Read – PCIe Request Dword 1**

Bit	Description
31:00	<b>Offset (OFFSET):</b> This field specifies the <del>least significant lower</del> 32-bits (i.e., bit 0 to bit 31) of the offset in bytes into the PCI BAR associated with the NVMe Controller at which the read begins.

**Figure 10: PCIe Memory Read – PCIe Request Dword 2**

Bit	Description
31:00	<b>Offset (OFFSET):</b> This field specifies the <del>upper</del> <b>most significant</b> 32-bits (i.e., bit 32 to bit 63) of the offset in bytes into the PCI BAR associated with the NVMe Controller at which the read begins.

*Modify a portion of Figure 133, Figure 134 and Figure 135 in section 7.6 as shown below:*

## 7.5 PCIe Memory Write

...

**Figure 11: PCIe Memory Write – PCIe Request Dword 0**

Bit	Description																
31:19	Reserved																
18:16	<b>Base Address Register (BAR):</b> This field specifies the PCI Base Address Register (BAR) of the memory space to be written. BARs are located beginning at 10h in PCI Configuration space and the value of this field specifies the starting offset of the associated BAR. For a 64-bit BAR, this field should correspond to the <del>lower</del> <b>least significant</b> 32-bits of the BAR. <table border="1" data-bbox="781 787 1101 1024"> <thead> <tr> <th>Value</th><th>BAR Offset</th></tr> </thead> <tbody> <tr><td>0h</td><td>10h</td></tr> <tr><td>1h</td><td>14h</td></tr> <tr><td>2h</td><td>18h</td></tr> <tr><td>3h</td><td>1Ch</td></tr> <tr><td>4h</td><td>20h</td></tr> <tr><td>5h</td><td>24h</td></tr> <tr><td>6h to 7h</td><td>Reserved</td></tr> </tbody> </table>	Value	BAR Offset	0h	10h	1h	14h	2h	18h	3h	1Ch	4h	20h	5h	24h	6h to 7h	Reserved
Value	BAR Offset																
0h	10h																
1h	14h																
2h	18h																
3h	1Ch																
4h	20h																
5h	24h																
6h to 7h	Reserved																
15:00	<b>Length (LENGTH):</b> This field specifies the number of bytes to be written.																

**Figure 12: PCIe Memory Write – PCIe Request Dword 1**

Bit	Description
31:00	<b>Offset (OFFSET):</b> This field specifies the <del>lower</del> <b>least significant</b> 32-bits (i.e., bit 0 to bit 31) of the offset in bytes into the PCI BAR associated with the NVMe Controller at which the write begins.

**Figure 13: PCIe Memory Write – PCIe Request Dword 2**

Bit	Description
31:00	<b>Offset (OFFSET):</b> This field specifies the <del>upper</del> <b>most significant</b> 32-bits (i.e., bit 32 to bit 63) of the offset in bytes into the PCI BAR associated with the NVMe Controller at which the write begins.

*Modify a portion of section 9.1 (Out-of-Band Operational Times) as shown below:*

## 9.1 Out-of-Band Operational Times

...

~~Request Messages are processed whenever MCTP access is supported on an interface (i.e., SMBus/I2C or PCIe). Although not recommended, an implementation may not support PCIe and SMBus/I2C MCTP accesses during a PCI Express conventional reset on any PCI Express port in the NVM Subsystem.~~ Although not recommended, an implementation may choose not to support processing of PCIe Commands that target a Controller in the NVM Subsystem that is in any of the following states:

*Modify a portion of section 9.3.1 as shown below:*

### 9.3.1 NVM Subsystem Reset

An NVM Subsystem Reset is initiated under the conditions outlined in the NVM Express 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~~ **executing** a Reset command.

*Modify a portion of section 9.3.2 as shown below:*

### 9.3.2 Controller Level Reset

...

The actions performed on a Controller Level Reset are outlined in the NVM Express specification. A Controller Level Reset has no effect on the Controller Management Interface associated with that Controller, the PCI Express port associated with that Controller, or a Management Endpoint associated with that port. A Controller Level Reset also does not stop the servicing of the Management Interface Command Set or NVM Express Admin Command Set commands that target that Controller (i.e., the NVM Express Admin Command Set is still available even though the NVMe Controller may be disabled or held in reset) or Control Primitives. A Controller Level Reset may affect PCIe Command Set commands ~~being processed~~ **executing** on that Controller (refer to section 0). If a PCIe Command is affected, then the command is completed with status PCIe Inaccessible.

*Modify a portion of section 9.3.3 as shown below:*

### 9.3.3 Management Endpoint Reset

...

In addition to these conditions, a Management Endpoint associated with a PCI Express port is reset when the PCI Express port is in any of the following states:

- A PCI Express conventional reset; or
- When the PCI Express link is down (i.e., not in the DL\_Active state).

When a Management Endpoint Reset is initiated, the state of that Management Endpoint is returned to its default condition and any commands associated with that Management Endpoint being processed are aborted.

A reset of a Management Endpoint in an NVM Subsystem ~~has no effect on~~ **shall not affect** any other Management Endpoint in the NVM Subsystem or any other NVM Subsystem entity. **Note that for implementations compliant to version 1.1 and earlier of this specification, during a PCI Express conventional reset of a PCIe Management Endpoint, MCTP accesses may not be supported on other PCIe or SMBus/I2C Management Endpoints in the NVM Subsystem.**

*Modify a portion of Appendix C to add a diagram that was in NVMe-MI 1.0 ECN 002 but was not applied to NVMe-MI 1.0a specification as shown below:*

## Appendix C – Example NVMe-MI Messages over SMBus/I2C

...

**Example 6:** This example shows an NVMe Storage Device sending an acknowledgement Response Message to the Replay Control Primitive and then sending a second Response Message that replays the previous Response Message from specified offset of zero. Note that the previous command is not reissued because that could return different data after having the Composite Controller Status cleared.

Start	BMC Addr	0	Ack	Protocol=MCTP	Ack	Length	Ack	SSD Addr	1	Ack	MCTP Version	Ack	BMC EID	Ack	SSD EID	Ack	flags, seq own, tag	Ack	Type= NVMe-MI	Ack	Cmd= Primitive	Ack	Rsvd	Ack	Rsvd	Ack
	20h			0Fh		11h		3Bh			01h		00h		00h		E4h		84h		80h		00h		00h	
Status= Success	Ack	Tag	Ack	CPSR Response	Ack	CPSR Rsvd	Ack	CRC32C LSB	Ack	CRC32C	Ack	CRC32C	Ack	CRC32C MSB	Ack	PEC	Ack	Stop								
	00h			45h		01h		00h		BDh		86h		02h		83h		94h								

Start	BMC Addr	0	Ack	Protocol=MCTP	Ack	Length	Ack	SSD Addr	1	Ack	MCTP Version	Ack	BMC EID	Ack	SSD EID	Ack	flags, seq own, tag	Ack	Type= NVMe-MI	Ack	Cmd= NVMe-MI	Ack	Rsvd	Ack	Rsvd	Ack	
	20h			0Fh		19h		3Bh			01h		00h		00h		F4h		84h		88h		00h		00h		
Status= Success	Ack	Rsvd	Ack	Rsvd	Ack	Rsvd	Ack	Rsvd	Ack	Subsystem Status	Ack	SMART Warnings	Ack	Composite Temp.	Ack	Percent Life Used	Ack	Ctrlr Stat LSB	Ack	Ctrlr Stat MSB	Ack	Rsvd	Ack	Rsvd	Ack		
00h		00h		00h		00h		00h		38h		FFh		1Eh		05h		01h		00h		00h		00h		00h	
CRC32C LSB	Ack	CRC32C	Ack	CRC32C	Ack	CRC32C MSB	Ack	PEC	Ack	Stop																	
C8h		3Bh		3Bh		57h		40h																			