LEGAL NOTICE:

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.

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

NVM Express® Technical Proposal (TP)

Technical Proposal ID	6038 Management Interface Miscellaneous Maintenance
Revision Date	2024.07.28
Builds on Specification(s)	NVM Express Management Interface Specification, Revision 1.2d NVM Express Base Specification, Revision 2.0d
References	ECN116 ECN120 TP4029a Power Loss Signal Support TP6027b Reset Behavior Enhancements TP6033a MCTP Packet Timing TP6035a Out-of-Band Asynchronous Events

Technical Proposal Author(s)

Name	**Company**
Judy Brock	Samsung
Austin Bolen	Dell
Myron Loewen	Solidigm

Technical Proposal Overview

This technical proposal addresses a number of miscellaneous maintenance issues. It will address various issues that impact interoperability between a Management Controller (e.g., a BMC) and a Management Endpoint (e.g., an NVMe Storage Device).

Revision History

Revision Date	Author	Change Description
2024.01.21	Judy Brock	• Initial draft.
2024.02.03	Judy Brock	• Incorporate Mike Allison feedback from 01/21 and MI WG feedback from 01/29 mtg. • Added new field Maximum Processing Suspended Time (MPST). • Clarified the operational states for Management Interface Command Set and PCIe Command Set commands with respect to controller disable and PCIe reset.

Revision Date	Author	Change Description
2024.03.18	Judy Brock	• Resolved various comments from 02/26 WG feedback. • Added new field Maximum Activation Without Reset Time (MAWRT) in Identify Controller indicating the full Firmware Commit with immediate activation time, and potentially time for activation after reset, not just the time the controller temporarily stops processing commands. • Added text from Myron related to boot failures/recovery actions.
2024.03.20	Judy Brock	• Added comments from 03/18 WG feedback.
2024.04.06	Judy Brock	• Added comments from 04/01 WG feedback, deleted old resolved comments, adjusted several field offsets in NVM Subsystem Element Descriptor due to decision to make Total NVM Capacity 12 bytes, not 16 bytes.
2024.04.22	Austin Bolen	• Added text to clarify the impact of disabling or resetting a Controller on out-of-band operations.
2024.04.27	Judy Brock	• Removed most of the new text in MI Command Set and PCIe Command Set sections that discusses the processing of MCTP commands while a controller is in disabled state or being held in reset by the host. This is now covered by Austin's new text in one place (section 8.1) for all MCTP command sets. • Accepted Formatting changes. • Deleted resolved comments.
2024.04.29	Judy Brock	• Cleaned up comments after 04/29 WG review. • Accepted blue text additions that have been reviewed.
2024.05.06	Austin Bolen	• Added remaining phase 2 functionality.
2024.05.08	Judy Brock	• Removed Total NVM Capacity (TNVMCAP) field. • Wordsmithing of text concerned with the processing of NVMe Admin Commands and NVMe-MI commands that target a controller that is being in a disabled state or being held in reset by the host. • Deleted resolved comments. • Accepted changes walked through in 05/06 WG mtg.
2024.05.15	Judy Brock	• Added Austin and Myron to the Authors list.
2024.05.18	Austin Bolen	• Phase 3 updates.
2024.05.20	Austin Bolen	• More Phase 3 updates.
2024.05.29	Austin Bolen	• Final phase 3 updates.
2024.05.30	Austin Bolen	• Updates from TWG review.
2024.07.02	Austin Bolen	• Address all member review feedback.
2024.07.15	Austin Bolen	• Address additional member review feedback.
2024.07.17	Austin Bolen	• Fixed a field name in a few places.
2024.07.21	Austin Bolen	• Fixed a reference error.
2024.07.27	Devin Allison	• Editorial updates.
2024.07.28	Austin Bolen	• Editorial updates.

Description for Changes Document for the NVM Express Management Interface Specification, Revision 1.2d

New Features/Feature Enhancements/Required Changes:

- Added the following new fields to the NVM Subsystem Element Descriptor data structure:
  - Management Endpoint Ready Independent of Media Timeout
  - Management Endpoint Ready With Media Timeout
  - **Incompatible Change:** Maximum Unresponsiveness Time
- Added the following new fields to the NVM Subsystem Port Descriptor data structure:
  - MCTP Support
- Added a Boot Failure Code field to the VPD that indicates failures to load or initialize the NVM Subsystem firmware
- Deleted the requirement for the Unique NVM Storage Device ID field of the UDID to be defined sequentially for each NVM Subsystem if there are multiple NVM Subsystems in an SMBus ARP-capable NVMe Storage Device or NVMe Enclosure
- Clarified the operational states with respect to controller disable and Controller Level Reset

Description for Changes Document for the NVM Express Base Specification, Revision 2.0d

New Features/Feature Enhancements/Required Changes:

- Clarified that for firmware activation without reset, the Firmware Commit command is completed after new firmware is activated
- Added the following new field to the Identify Controller data structure:
  - Maximum Processing Time for Firmware Activation Without Reset (MPTFAWR)

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 or reader
Orange:	Text is pulled in from a referenced Technical Proposal
~~Orange Strikethrough~~:	Text is deleted from a referenced Technical Proposal

Description of Specification Changes for NVM Express Management Interface Specification 1.2d

2.2 SMBus/I2C

…

<Note to Editor: The following sentence from the NVM Express Management Interface Specification, Revision 1.2d was partially modified by ECN120 and is now being completely deleted by this TP.>

~~If there are multiple NVM Subsystems in an SMBus ARP-capable NVMe Storage Device or NVMe Enclosure, then for each NVM Subsystem the Unique NVM Storage value of the UDID Device ID fields of the UDID shall be sequential incremented by one for each NVM Subsystem.~~ If the Upstream Connector has an SMBus/I2C port, then the FRU Information Device associated with that connector shall be present directly on the SMBus/I2C channel connected to the Upstream Connector.

4.1.2.3 More Processing Required Response

A More Processing Required Response shall be returned when the Management Endpoint requires more than the maximum Request-To-Response Time (refer to section 4.2.2.1) to complete the Process state of the Command Message unless otherwise specified (e.g., the More Processing Required Response may be discarded under certain conditions as described in section 4.2 or the Request Message may be discarded under certain conditions as described in section 8.1). If a More Processing Required Response is returned, then the Management Endpoint shall start to transmit the More Processing Required Response before the maximum Request-To-Response Time is exceeded unless otherwise specified (refer to section 4.2.2.1.1). If a Get State Control Primitive is processed while the Management Endpoint is transmitting the More Processing Required Response, then the Management Endpoint shall indicate a value of 2h (i.e., Process) in the Slot Command Servicing State field in the Get State Control Primitive Response Message (refer to Figure 42).

…

5 Management Interface Command Set

The Management Interface Command Set defines the Command Messages that may be submitted by a Requester when the NMIMT value is set to NVMe-MI Command. The Management Interface Command Set is applicable to both the out-of-band mechanism and the in-band tunneling mechanism. The processing of commands in the Management Interface Command Set may be affected by the Command and Feature Lockdown feature (refer to the NVM Express Base Specification).

The servicing of any NVMe-MI Command over the out-of-band mechanism shall be independent of and not affected by any one or more Controllers in the NVM Subsystem being disabled or being reset by a Controller Level Reset unless the Management Endpoint servicing the NVMe-MI Command is reset (e.g., due to an NVM Subsystem Reset or due to a PCIe Reset of the PCIe VDM Management Endpoint servicing the NVMe-MI Command). Refer to section 8.1 for more details.

The NVMe-MI Message structure with fields that are common to all NVMe-MI Messages is defined in section 3.1.

…

6 NVM Express Admin Command Set

...

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 NVMe commands (e.g., Identify). Coordination between a Management Controller and host is outside the scope of this specification.

The servicing of any NVMe Admin Command~~s~~ over the out-of-band mechanism ~~that targets a Controller in the NVM Subsystem~~ shall be independent of and not affected by any one or more ~~may target a~~ c~~C~~ontrollers ~~that is~~ in the NVM Subsystem being disabled or being ~~held in~~ reset by a Controller Level Reset unless the Management Endpoint servicing the NVMe Admin Command is reset (e.g., due to an NVM Subsystem Reset or due to a PCIe Reset of the PCIe VDM Management Endpoint servicing the NVMe Admin Command) ~~the host~~. ~~When this occurs, the NVMe Admin Command is processed normally.~~ Refer to section 8.1 for more details.

The Request Message format for NVMe Admin Commands is shown in Figure 117 and is described Figure 118.

...

7 PCIe Command Set (Optional)

…

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

The servicing of any PCIe Command shall be independent of and not affected by any one or more Controllers in the NVM Subsystem being:

- disabled; or
- reset by a Controller Level Reset unless the Management Endpoint servicing the PCIe Command is reset (e.g., due to an NVM Subsystem Reset or due to a PCIe Reset of the PCIe VDM Management Endpoint servicing the PCIe Command), unless otherwise specified.

  Refer to section 8.1 for more details.

The Request Message format for PCIe Commands is shown in Figure 128 and described in Figure 129.

…

<Note to Editor: Change all occurrences of “SMBus/I2C Mux” in this TP to “2-Wire Mux” in the integration specification.>

8.1 Out-of-Band Operational Times

In the out-of-band mechanism, the ability of a Management Endpoint to receive and process Request Messages outlined in this specification is dependent on the state of the Management Endpoint. This section enumerates Management Endpoint, FRU Information Device, and SMBus/I2C Mux operational times and the operations supported in each of these operational times.

The NVM Subsystem power state is defined by the state of main power and auxiliary power. Main power consists of one or more voltage rails as defined by form factor. When main power consists of multiple voltage rails, main power is considered “on” when power is good on all main voltage rails. Auxiliary power is optionally supported by a form factor and enables SMBus/I2C communications in the absence of main power. Only the Powered On and Powered Off states are applicable in form factors and platforms that do not support auxiliary power. Figure 147 defines the power states of a Management Endpoint. Note that auxiliary power is described from the perspective of the NVM Subsystem and could be provided by any appropriate power rail in a host platform.

The operations supported in each NVM Subsystem power state are summarized in Figure 147. ~~VPD~~ SMBus/I2C VPD access consists of processing read operations to the FRU Information Device. SMBus/I2C Mux access consists of processing operations to the SMBus/I2C Mux. SMBus/I2C MCTP access consists of processing and responding to ~~MCTP m~~NVMe-MI Messages on the NVM Subsystem SMBus/I2C port. PCIe MCTP access consists of processing and responding to ~~MCTP m~~NVMe-MI Messages issued on any NVM Subsystem PCIe port. The behavior of an operation that is “Not Supported” in Figure 147 is undefined.

Figure 147: Operations Supported During NVM Subsystem Power States

Operation[4, 5]	Powered Off -All Power Rails Off	Powered On -All Power Rails On	Auxiliary Power Only[2] -Main Power Off -Auxiliary Power On	Main Power Only[2] -Main Power On -Auxiliary Power Off
SMBus/I2C VPD and SMBus/I2C Mux Access	Not Supported	Supported[3]	Supported[3]	Implementation Specific
SMBus/I2C MCTP Access	Not Supported	Supported[3]	Optional[1]	Implementation Specific[3]
PCIe MCTP Access	Not Supported	Supported[3]	Not Supported	Supported[3]

Notes:

1. An implementation that supports SMBus/I2C MCTP Access during Auxiliary Power may support a subset of commands during this power state. The commands that are supported are implementation specific.
2. Auxiliary Power Only and Main Power Only columns are not applicable to form factors that do not define Auxiliary power.
3. For interactions with Power Loss Signaling processing, refer to section 8.1.1.
4. An SMBus Reset may prevent access as described in section 8.3.4. A PCIe Reset may prevent access as described in this section, section 8.3.2, and section 8.3.5.
5. Firmware activation may impact access as described in section 8.3.2.

~~When~~ Within 1 s after an NVM Subsystem transitions from a power state in which accesses are not supported to one where accesses are supported, ~~accesses shall be processed 1 s after entering the power state in which are supported~~ the accesses listed in Figure 147 are operational.

Once operational, SMBus/I2C VPD accesses and SMBus/I2C Mux accesses shall be processed unless otherwise noted. SMBus/I2C VPD accesses SMBus/I2C Mux accesses are permitted to be unsupported once operational during the following cases:

- a firmware activation without reset (i.e., the Commit Action field in the Firmware Commit command is set to 011b) is being processed (refer to the NVM Express Base Specification);
- a VPD Read command is being processed;
- a VPD Write command is being processed; or
- while updating the Boot Failure Code field or Common Header Checksum field due to a boot failure detected while operational (refer to Figure 167).

If SMBus/I2C VPD accesses or SMBus/I2C Mux accesses are not supported once operational, then the FRU Information Device or SMBus/I2C Mux, respectively, shall be hidden from the Management Controller on the SMBus/I2C port (e.g., by detaching the FRU Information Device or SMBus/I2C Mux from the Management Controller-facing SMBus/I2C port). While the FRU Information Device or SMBus/I2C Mux is hidden from the Management Controller, SMBus/I2C VPD accesses or SMBus/I2C Mux accesses from the Management Controller shall be NACKed.

<Note to Editor: Change the occurrences of “SMBus/I2C port” in the prior paragraph to “2-Wire port” in the integration specification.>

If SMBus/I2C VPD accesses or SMBus/I2C Mux accesses are not supported once operational, then it is strongly recommended that the amount of time that accesses are unsupported be minimized. If these accesses are not supported once operational, then a Management Controller may time out and report the NVM Subsystem as failed.

Once operational, all other accesses other than SMBus/I2C VPD accesses and SMBus/I2C Mux accesses should be processed~~ 1 s after entering the power state in which accesses are supported~~. For example, an SMBus/I2C VPD or SMBus/I2C Mux ~~MCTP~~ access issued 1 s after transitioning from a "Powered Off" to a "Main Power" state is guaranteed to be processed and an MCTP access should be processed. The behavior of accesses prior to this 1 s time interval is undefined.

If a Request Message is received greater than or equal to 1 s after entering a power state in which MCTP accesses are supported from a power state in which MCTP access are not supported and the Management Endpoint is not ready to process the Request Message, then the Management Endpoint should return a More Processing Required Response (refer to section 4.1.2.3). Upon entering a power state in which MCTP accesses are supported from a power state in which MCTP access are not supported, the maximum amount of time a Management Endpoint is ready to start processing a Request Message that:

- does not require media access is indicated by the Management Endpoint Ready Independent of Media Timeout (MERIMTO) field; and
- requires media access is indicated by the Management Endpoint Ready With Media Timeout (MERWMTO) field.

In certain cases, upon entering a power state in which MCTP accesses are supported from a power state in which MCTP access are not supported, the Management Endpoint may be unresponsive and unable to service a Request Message or return any Response Message, including a More Processing Required Response Message (e.g., due to activating a firmware image or due to executing code in a security module that prohibits execution of code outside the security module that is required to process the Request Message). Upon entering a power state in which MCTP accesses are supported from a power state in which MCTP access are not supported, the Management Endpoint shall not be unresponsive for more than the amount of time indicated by the Maximum Unresponsive Time field. Any Request Message received during the time a Management Endpoint is unresponsive shall be silently discarded.

Note that it is strongly recommended that the amount of time a Management Endpoint is unresponsive be minimized. If a Management Endpoint is unresponsive, then Management Controllers compliant with revisions of this specification prior to the definition of the Maximum Unresponsive Time field may time out and report the NVM Subsystem as failed.

When transitioning between power states in which accesses are supported in both states (i.e., the state before and after the transition), there is no interruption in access processing (i.e., accesses are processed prior to the state transition, during the state transition, and immediately after entering the new power state).

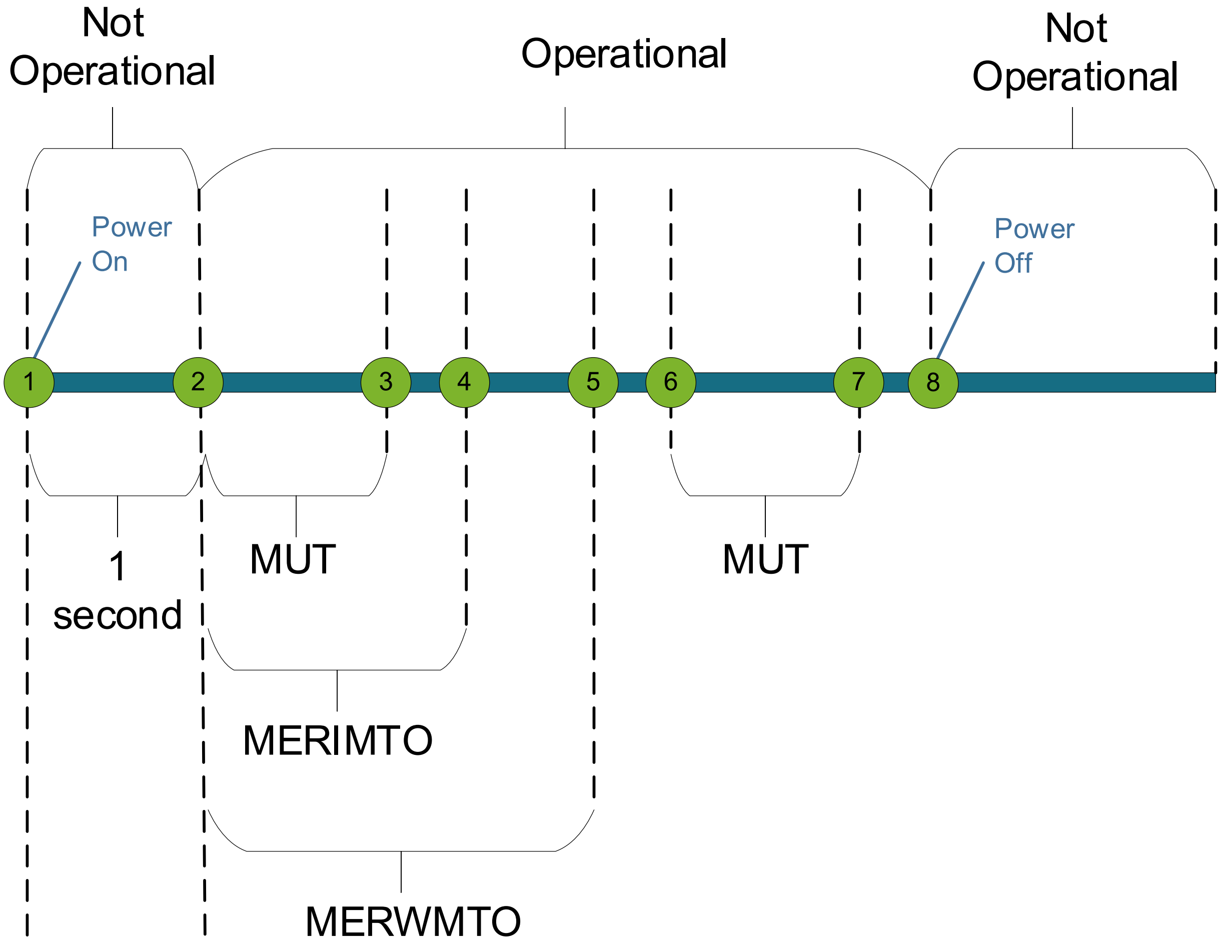
An example operational timing diagram is shown in Figure TBD2. In this example, there are no shutdowns of any Controller or of the NVM Subsystem. And in this example, the value in the MUT field is smaller than the value in the MERIMTO field which is smaller than the value in the MERWMTO field, but this ordering is not required. The sequence of events in Figure TBD2 is as follows:

1. The NVM Subsystem transitions from a power state in which accesses to a Management Endpoint, FRU Information Device, and/or SMBus/I2C Mux are not supported to one where accesses are supported. After entering the power state in which accesses are supported, any Management Endpoint, FRU Information Device, or SMBus/I2C Mux in the NVM Subsystem is permitted to be non-operational for a maximum time of 1 s. While non-operational, none of the operations in Figure 147 are supported and if a Management Controller performs any of those operations, then the results are undefined.
2. Within 1 s after transitioning from a power state in which accesses are not supported to one where accesses are supported, SMBus/I2C VPD and SMBus/I2C Mux accesses are processed, unless otherwise specified, and MCTP accesses should be processed.
    a. If the Management Endpoint is not ready to process an MCTP access that does not require media access, then the Management Endpoint is permitted to return a More Processing Required Response for up to the amount of time indicated by the MERIMTO field.
    b. If the Management Endpoint is not ready to process an MCTP access that requires media access, then the Management Endpoint is permitted to return a More Processing Required Response for up to the amount of time indicated by the MERWMTO field.

   c. If the Management Endpoint is unresponsive and unable to service a Request Message or return any Response Message, including a More Processing Required Response Message (e.g., due to activating a firmware image or due to executing code in a security module that prohibits execution of code outside the security module that is required to process the Request Message), then the Management Endpoint discards any received Request Messages.

3. Within the amount of time indicated by the MUT field since entering the operational state, the Management Endpoint is no longer permitted to be unresponsive to MCTP accesses.
4. Within the amount of time indicated by the MERIMTO field since entering the operational state, the Management Endpoint is ready to process Request Messages that do not require access to media.
5. Within the amount of time indicated by the MERWMTO field since entering the operational state, the Management Endpoint is ready to process Request Messages that require access to media.
6. While operational, if the Management Endpoint is unresponsive and unable to service a Request Message or return any Response Message, including a More Processing Required Response Message (e.g., due to activating a firmware image without reset), then the Management Endpoint discards any received Request Messages. While operational, if SMBus/I2C VPD accesses or SMBus/I2C Mux accesses are not supported, then the SMBus/I2C VPD accesses or SMBus/I2C Mux accesses are NACKed.
7. While operational, the Management Endpoint is unresponsive to MCTP accesses, SMBus/I2C VPD accesses, and SMBus/I2C Mux accesses for no longer than the amount of time indicated by the MUT field.
8. The NVM Subsystem transitions from a power state in which accesses to a Management Endpoint, FRU Information Device, and/or SMBus/I2C Mux are supported to one where accesses are not supported. The behavior of any access from the Management Controller that is not supported in the current power state is undefined.

Figure TBD2 Operational Time Example Timing Diagram



Key:
MUT: Maximum Unresponsiveness Time field value
MERIMTO: Management Endpoint Read Independent of Media Timeout field value
MERWMTO: Management Endpoint Read With Media Timeout field value

8.1.1 Controller Disable and Reset Interactions

<Note to Editor: In the following three paragraphs, change every occurrence of “SMBus/I2C” to “2-Wire” in the integration specification.>

The enable/disable state of any Controller in the NVM Subsystem (refer to the CC.EN bit in the NVM Express Base Specification) shall have no effect on any operations in Figure 147. For example, in a power state where SMBus/I2C MCTP access or PCIe MCTP access is supported, it is not an error to submit a Request Message to a disabled Controller, including Request Messages such as NVMe Admin Commands that target the disabled Controller (e.g., the Management Endpoint processes an NVMe Admin Command the same as if the Controller were enabled). Likewise, disabling a Controller while one or more operations from Figure 147 are being serviced, including Request Messages targeting the Controller being disabled, shall have no effect on the servicing of those operations.

<Note to Editor: Section 8.3.TBD in the following paragraph is referring to the “PCIe Reset” section added by TP6027b.>

Controller Level Resets shall have no impact on any operations in Figure 147 except for the impacts to PCIe MCTP accesses to PCIe VDM Management Endpoints due to PCIe Resets (i.e., Conventional Reset or Function Level Reset) described in section 8.3.TBD and the impacts to PCIe Command processing described in this section. For example, in a power state where SMBus/I2C MCTP access is supported and unless otherwise specified, it is not an error to submit a Request Message via MCTP over SMBus/I2C to a Controller in a PCIe Reset asserted state including Request Messages that target the Controller in the PCIe Reset asserted state such as NVMe Admin Commands (e.g., the Management Endpoint processes an NVMe Admin Command the same as if the Controller was not in a PCIe Reset asserted state). Likewise, unless otherwise specified, asserting Controller Level Reset to a Controller while one or more Request Messages are being serviced, including Request Messages targeting the Controller being reset, shall have no effect on the servicing of any Request Message.

Unless otherwise specified, if the NVM Subsystem is in a power state in which an operation in Figure 147 is supported, then the Management Endpoint shall complete any steps required to be able to successfully handle the operation and then handle the operation. For example, if an NVMe Admin Command targeting a Controller that is in normal operation (i.e., the value of the CSTS.SHST field is cleared to 00b) and has not been shut down requires media access and media has not been initialized, then the Management Endpoint shall initialize media and then the NVMe Admin Command shall be processed. If an operation could be processed successfully but the Management Endpoint instead returns an error (e.g., an Error Response for any Command Message or an error status code in the Status field in CQEDW3 in an NVMe Admin Command Response) due to any reason including not attempting to complete any steps necessary to successfully complete the operation (e.g., initializing media), then the Management Controller may erroneously flag the NVM Subsystem as failed.

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:[1~~2~~]

- Controller Level Reset that does not reset the Management Endpoint servicing the PCIe Command;
- SR-IOV ~~v~~Virtual ~~f~~Function is not enabled;
- ~~D~~during any type of PCI Express Conventional Reset, for PCIe Commands received via MCTP over SMBus/I2C;

<Note to Editor: Change “over SMBus/I2C” in the prior bullet point to “over 2-Wire” in the integration specification.>

- ~~D~~during a PCI Express Function Level Reset (FLR);
- ~~W~~when the PCI Express Function is in a non-D0 power D-state; or
- ~~W~~when the PCI Express link is down (i.e., not in the DL_Active state).

If a PCIe Command is received that targets a Controller in one of these states and the implementation does not support processing of PCIe Commands in that state, then the PCIe command is completed with status PCIe Inaccessible. ~~Processing of supported PCIe Commands is required~~ The Controller shall not complete PCIe Commands with a status of PCIe Inaccessible in all other Controller states.

If a PCIe Command is received that targets a Controller whose corresponding PCIe link is in a low power state (i.e., PCIe ASPM), then processing of the command may cause the link to temporarily exit the low power state.

[1~~2~~] A Management Controller ~~shall~~ should only send these commands using SMBus/I2C or another PCIe port since the link associated with the PCIe port and Controller is down in these states.

8.1.2~~1~~ Power Loss Signaling Interactions

A Controller which responds to a Power Loss Signaling notification performs either Forced Quiescence Processing or Emergency Power Fail Processing (refer to the Power Loss Signaling section of the NVM Express Base Specification).

If one or more Controllers in an NVM Subsystem are in the:

a) FQ Processing state;
b) FQ Complete state;
c) EPF Processing Port Enabled state;
d) EPF Complete Port Enabled state;
e) EPF Processing Port Disabled state; or
f) EPF Complete Port Disabled state,

then:

a) all Command Slots in ~~the~~ all Management Endpoints in that NVM Subsystem should:
    a) behave as if an implicit Abort Control Primitive (refer to section 4.2.1.3) was received with the exception that the Management Endpoint shall not transmit the Abort Control Primitive Response Messages; and
    b) drop (silently discard) Control Primitives;
b) access to an SMBus/I2C VPD in that NVM Subsystem may or may not be supported; and
c) access to the SMBus/I2C Mux in that NVM Subsystem may or may not be supported.

When all Controllers in an NVM Subsystem have transitioned out of the FQ Complete state because the PLN variable transitioned to Deasserted, then:

a) the Management Endpoint shall service Request Messages;
b) access to the SMBus/I2C VPD shall be supported; and
c) access to the SMBus/I2C Mux shall be supported.

8.2.1 Common Header

The fields that make up the VPD Common Header are shown in Figure 150.

Figure 150: Common Header

Bytes	Factory Default	Description
0	01h	**IPMI Format Version Number (IPMIVER):** This field indicates the IPMI Format Version.
1	Impl Spec	**Internal Use Area Starting Offset (IUAOFF):** This field indicates the starting offset in multiples of 8 bytes for the Internal Use Area. A value of 0h may be used to indicate the Internal Use Area is not present.
2	Impl Spec	**Chassis Info Area Starting Offset (CIAOFF):** This field indicates the starting offset in multiples of 8 bytes for the Chassis Info Area. A value of 0h may be used to indicate the Chassis Info Area is not present.
3	Impl Spec	**Board Info Area Starting Offset (BIAOFF):** This field indicates the starting offset in multiples of 8 bytes for the Board Info Area. A value of 0h may be used to indicate the Board Info Area is not present.
4	Impl Spec	**Product Info Area Starting Offset (PIAOFF):** This field indicates the starting offset in multiples of 8 bytes for the Product Info Area. A value of 0h may be used to indicate the Product Info Area is not present.
5	Impl Spec	**MultiRecord Info Area Starting Offset (MRIOFF):** This field indicates the starting offset in multiples of 8 bytes for the MultiRecord Info Area.

Figure 150: Common Header

<table>
<tr><th>Bytes</th><th>Factory Default</th><th>Description</th></tr>
<tr><td>6</td><td>00h</td><td><b>Boot Failure Code (BFC):</b> If the Boot Failure Code Support bit (refer to Figure 174) is set to ‘1’, then this field shall indicate the applicable boot failure code from the following table. A boot failure is a failure to load or initialize the NVM Subsystem firmware. If this field is updated, then the Common Header Checksum field shall be updated.
If the Boot Failure Code Support bit is cleared to ‘0’, then this field is reserved.
<table>
<tr><th>Boot Failure Code</th><th>Description</th></tr>
<tr><td>0h</td><td>No boot failure has occurred</td></tr>
<tr><td>1h</td><td>Unrecoverable Hardware Issue</td></tr>
<tr><td>2h</td><td>Self-test Failure</td></tr>
<tr><td>4h</td><td>Corrupted Critical Data</td></tr>
<tr><td>6h</td><td>Corrupted Key Manifest</td></tr>
<tr><td>8h</td><td>Corrupted Firmware Image</td></tr>
<tr><td>Ah</td><td>Corrupted Security Data</td></tr>
<tr><td>Ch</td><td>Corrupted Recovery Firmware</td></tr>
</table>
~~Reserved~~</td></tr>
<tr><td>7</td><td>Impl Spec</td><td><b>Common Header Checksum (CHCHK):</b> Checksum computed over byte 0 to byte 6. The checksum is computed by adding the 8-bit value of the bytes modulo 256 and then taking the 2’s complement of this sum. When the checksum and the sum of the bytes module 256 are added, the result should be 0h.</td></tr>
</table>

8.2.4 NVMe PCIe Port MultiRecord Area

…

Figure 154: NVMe PCIe Port MultiRecord Area

<table>
<tr><th>Bytes</th><th>Factory Default</th><th>Description</th></tr>
<tr><td colspan="3">…</td></tr>
<tr><td>10</td><td>Impl Spec</td><td><b>MCTP Support:</b> This field ~~contains~~ indicates a bit vector that specifies the level of support for the NVMe Management Interface.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>7:1</td><td>Reserved</td></tr>
<tr><td>0</td><td>If MCTP-based NVMe-MI Messages are supported on this PCIe port, then this bit shall be set to ‘1’; otherwise, this bit shall be cleared to ‘0’. ~~If this bit is set to ‘1’, then MCTP-based management commands are supported on the PCIe port. If this bit is cleared to ‘0’, then MCTP-based management commands are not supported on the PCIe port.~~</td></tr>
</table></td></tr>
<tr><td colspan="3">…</td></tr>
</table>

8.2.5.7 NVM Subsystem Element Descriptor

The NVM Subsystem Element Descriptor is shown in Figure 189 and is used to describe an NVM Subsystem contained in the NVMe Storage Device.

Figure 172: NVM Subsystem Element Descriptor

Bytes[1]	Factory Default	Description
00	07h	**Type:** This field indicates the type of the Element Descriptor. This field shall be set to the NVM Subsystem Element Descriptor Type (i.e., 7h). Refer to Figure 175.

Figure 172: NVM Subsystem Element Descriptor

Bytes[1]	**Factory Default**	**Description**
01	01~~0~~h	**Revision:** This field indicates the revision of the NVM Subsystem Element Descriptor. This field shall be ~~cleared to 0h~~ set to 1h.
02	Impl Spec	**Length:** This field indicates the length of the NVM Subsystem Element Descriptor in bytes.
…	**…**	**…**
05	Impl Spec	**NVM Subsystem Port Descriptor Count (NVMSPDC):** This field indicates the number of NVM Subsystem Port Descriptors associated with the NVM Subsystem. The permitted range of values is 1 to 64.
~~Impl Spec~~ 6+M-1:6	Impl Spec	**NVM Subsystem Port Descriptor 0:** This field contains the NVM Subsystem Port Descriptor associated with the first NVM Subsystem port.
~~Impl Spec~~ 6+(2*M)-1: 6+M	Impl Spec	**NVM Subsystem Port Descriptor 1:** This field contains the NVM Subsystem Port Descriptor associated with the second NVM Subsystem port if NVM Subsystem Port Descriptor Count field is greater than one, otherwise this field is not present.
…	…	…
~~Impl Spec~~ 6+(N*M)-1: 6+((N-1)*M)	Impl Spec	**NVM Subsystem Port Descriptor N:** This field contains the NVM Subsystem Port Descriptor associated with the last NVM Subsystem port if the NVM Subsystem Port Descriptor Count field is greater than 2h; otherwise, this field is not present. N is equal to the NVMSPDC field minus 1h.
X+1:X	Impl Spec	**Management Endpoint Ready Independent of Media Timeout (MERIM):** This field shall indicate the maximum time in 100 ms units required by a Management Endpoint after entering a power state in which accesses are supported from a power state in which accesses are not supported (refer to section 8.1), to be ready to start processing a Request Message that does not require media access. A value of 0h indicates that no time is indicated. This field shall not be cleared to 0h on implementations that are compliant with revisions later than 1.2 of this specification that support SMBus/I2C VPD accesses. <Note to Editor: Change "revisions later than 1.2" in the prior sentence to "revision 2.0 or later" in the integration specification.>
X+3:X+2	Impl Spec	**Management Endpoint Ready With Media Timeout (MERWMTO):** This field shall indicate the estimated maximum time in 100 ms units required by a Management Endpoint after entering a power state in which accesses are supported from a power state in which accesses are not supported (refer to section 8.1), to be ready to start processing a Request Message that requires media access. A value of 0h indicates that no time is indicated. This field shall not be cleared to 0h on implementations that are compliant with revisions later than 1.2 of this specification that support SMBus/I2C VPD accesses. <Note to Editor: Change "revisions later than 1.2" in the prior sentence to "revision 2.0 or later" in the integration specification.>

Figure 172: NVM Subsystem Element Descriptor

Bytes[1]	Factory Default	Description
X+5:X+4	Impl Spec	**Maximum Unresponsive Time (MUT):** This field shall indicate the estimated maximum time in 100 ms units once operational (refer to section 8.1) that: • the Management Endpoint is permitted to be unable to service Request Messages or AEMs for any reason (e.g., due to activating a firmware image or due to executing code in a security module that prohibits execution of code outside the security module that is required to process a Request Message); and • SMBus/I2C VPD accesses or SMBus/I2C Mux accesses are permitted to be unsupported due to the conditions listed in section 8.1. The ability for accesses to the Management Endpoint, SMBus/I2C VPD, and SMBus/I2C Mux to be unresponsive are independent (e.g., an NVM Subsystem may be in a state where accesses to the Management Endpoint are unresponsive and accesses to the SMBus/I2C VPD are responsive or vice versa). This field shall not include the time to ready for the Management Endpoint indicated by the MERIMTO and MERWMTO fields. A value of 0h indicates that no time is indicated. This field shall not be cleared to 0h on implementations that are compliant with revisions later than 1.2 of this specification that support SMBus/I2C VPD accesses. <Note to Editor: Change “revisions later than 1.2” in the prior sentence to “revision 2.0 or later” in the integration specification.>

Notes:

1. When used in this column:
   - N is equal to the value of the NVMSPDC;
   - M is equal to the value of the Length field in the NVM Subsystem Port Descriptor data structure (refer to Figure 173); and
   - X is equal to the starting byte offset of the MERIMTO field (i.e., 6+(NVMSPDC*M)).

Each upstream port is described by an NVM Subsystem Port Descriptor as shown in Figure 173. ~~It~~ The NVM Subsystem Port Descriptor describes the PCIe port’s supported PCIe link speeds, PCIe max link width, RefClk capabilities, ~~and~~ PCIe Port Identifier, and MCTP support. Each NVM Subsystem Port Descriptor should be the child of exactly one parent Element Descriptor.

Figure 173: NVM Subsystem Port Descriptor

<table>
<tr><th>Bytes</th><th>Factory Default</th><th>Description</th></tr>
<tr><td>00</td><td>00h</td><td><b>Type:</b> This field indicates the type of an NVM Subsystem Port Descriptor. This field shall be cleared to 0h.</td></tr>
<tr><td>01</td><td>Impl Spec</td><td><b>Length:</b> This field indicates the length of the NVM Subsystem Port Descriptor in bytes.</td></tr>
<tr><td>02</td><td>Impl Spec</td><td><b>PCIe Link Speed:</b> This field indicates a bit vector of link speeds supported by the PCIe port.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7:5</td><td>Reserved</td></tr>
<tr><td>4</td><td>Set to ‘1’ if the PCIe link supports 32.0 GT/s, otherwise cleared to ‘0’.</td></tr>
<tr><td>3</td><td>Set to ‘1’ if the PCIe link supports 16.0 GT/s, otherwise cleared to ‘0’.</td></tr>
<tr><td>2</td><td>Set to ‘1’ if the PCIe link supports 8.0 GT/s, otherwise cleared to ‘0’.</td></tr>
<tr><td>1</td><td>Set to ‘1’ if the PCIe link supports 5.0 GT/s, otherwise cleared to ‘0’.</td></tr>
<tr><td>0</td><td>Set to ‘1’ if the PCIe link supports 2.5 GT/s, otherwise cleared to ‘0’.</td></tr>
</table></td></tr>
</table>

Figure 173: NVM Subsystem Port Descriptor

<table>
<tr><th>Bytes</th><th>Factory Default</th><th>Description</th></tr>
<tr><td>03</td><td>Impl Spec</td><td><b>PCIe Maximum Link Width:</b> The maximum PCIe link width for this NVM Subsystem port.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>Reserved</td></tr>
<tr><td>1</td><td>PCIe x1</td></tr>
<tr><td>2</td><td>PCIe x2</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
<tr><td>4</td><td>PCIe x4</td></tr>
<tr><td>5 to 7</td><td>Reserved</td></tr>
<tr><td>8</td><td>PCIe x8</td></tr>
<tr><td>9 to 11</td><td>Reserved</td></tr>
<tr><td>12</td><td>PCIe x12</td></tr>
<tr><td>13 to 15</td><td>Reserved</td></tr>
<tr><td>16</td><td>PCIe x16</td></tr>
<tr><td>17 to 31</td><td>Reserved</td></tr>
<tr><td>32</td><td>PCIe x32</td></tr>
<tr><td>33 to 255</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>04</td><td>Impl Spec</td><td><b>RefClk Capability:</b> This field contains a bit vector that specifies the PCIe clocking modes supported by the port.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7:4</td><td>Reserved</td></tr>
<tr><td>3</td><td>Set to ‘1’ if the device automatically uses RefClk if provided and otherwise uses SRIS, otherwise cleared to ‘0’.</td></tr>
<tr><td>2</td><td>Set to ‘1’ if the PCIe link supports Separate RefClk with SSC (SRIS), otherwise cleared to ‘0’.</td></tr>
<tr><td>1</td><td>Set to ‘1’ if the PCIe link supports Separate RefClk with no SSC (SRNS), otherwise cleared to ‘0’.</td></tr>
<tr><td>0</td><td>Set to ‘1’ if the PCIe link supports common RefClk, otherwise cleared to ‘0’.</td></tr>
</table></td></tr>
<tr><td>05</td><td>Impl Spec</td><td><b>Port Identifier:</b> This field contains the NVMe-MI Port Identifier associated with this port.</td></tr>
<tr><td>6</td><td>Impl Spec</td><td><b>MCTP Support (MCTPS):</b> This field indicates a bit vector that specifies the level of support for the NVMe Management Interface.
<table>
<tr><th>Bits</th><th>Definition</th></tr>
<tr><td>7:1</td><td>Reserved</td></tr>
<tr><td>0</td><td><b>MCTP PCIe VDM Support (MCTPPCIEVS):</b> If MCTP-based NVMe-MI Messages are supported on this PCIe port, then this bit shall be set to ‘1’; otherwise, this bit shall be cleared to ‘0’.</td></tr>
</table></td></tr>
</table>

8.2.5.8 FRU Information Device Element Descriptor

The FRU Information Device Element Descriptor is shown in Figure 174 and is used to describe a FRU Information Device contained in the NVMe Storage Device.

Figure 174 : 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 (i.e., 8). Refer to Figure 158.
01	00h	**Revision:** This field indicates the revision of the FRU Information Device Element Descriptor. This field shall be set to 1h ~~cleared to 0h~~.
02	06h	**Length:** This field indicates the length of the FRU Information Device Element Descriptor in bytes.

Figure 174 : FRU Information Device Element Descriptor

<table>
<tr><th>Byte Offset</th><th>Factory Default</th><th>Description</th></tr>
<tr><td>03</td><td>A6h/A7h or 0h for NVM Storage Devices A4h/A5h or 0h for Carriers</td><td><b>SMBus/I2C Address Info:</b> If the NVMe Storage Device contains an SMBus/I2C port, then this field indicates the default SMBus/I2C addressing per the table below; else, this field shall be cleared to 0h.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7:1</td><td><b>SMBus/I2C Address:</b> This field contains the 7-bit SMBus/I2C address. Refer to Figure 16 for requirements.</td></tr>
<tr><td>0</td><td><b>ARP Capable:</b> If this bit is set to ‘1’, then SMBus ARP is supported. If this bit is cleared to ‘0’, then SMBus ARP is not supported. Refer to the SMBus Specification for additional details.</td></tr>
</table></td></tr>
<tr><td>04</td><td>Impl Spec</td><td><b>SMBus/I2C Capabilities:</b> If the NVM Storage Device contains an SMBus/I2C port, then this field indicates the SMBus/I2C capabilities per the table below; else, this field shall be cleared to 0h.
<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>7</td><td><b>Reset:</b> If this bit is set to ‘1’, then all of the SMBus/I2C reset mechanisms are supported as defined by the specification for the Form Factor in the Host Connector Element Descriptor.
If this bit is cleared to ‘0’, then the FRU Information 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.</td></tr>
<tr><td>6</td><td><b>I2C Writes Allowed:</b> If this bit is set to ‘1’, then the FRU Information Device is allowed to be written using an I2C Write operation.
If this bit is cleared to ‘0’, then the FRU Information Device is not allowed to be written using an I2C Write operation.</td></tr>
<tr><td>5</td><td><b>Boot Failure Code Support (BFCS):</b> If SMBus/I2C VPD accesses are supported (refer to section 8.1) and the Boot Failure Code field (refer to Figure 167) is supported, then this bit shall be set to ‘1’; otherwise, this bit shall be cleared to ‘0’.
This bit shall not be cleared to ‘0’ on implementations that are compliant with revisions later than 1.2 of this specification that support SMBus/I2C VPD accesses.
<Note to Editor: Change “revisions later than 1.2” in the prior sentence to “revision 2.0 or later” in the integration specification.&gt;</td></tr>
<tr><td>~~5~~4:2</td><td>Reserved</td></tr>
<tr><td>1:0</td><td><b>Maximum Speed</b>: This field is set to the highest supported SMBus/I2C clock speed supported by the FRU Information Device.
<table>
<tr><th>Value</th><th>Description</th></tr>
<tr><td>0</td><td>100 kHz</td></tr>
<tr><td>1</td><td>400 kHz</td></tr>
<tr><td>2</td><td>1 MHz</td></tr>
<tr><td>3</td><td>Reserved</td></tr>
</table></td></tr>
</table></td></tr>
<tr><td>05</td><td>8h to 0Ch inclusive</td><td><b>Maximum FRU Information Device Size:</b> The maximum size of the FRU Information Device is 2^N bytes where N is the value in this field (e.g., a value of 8 in this field indicates a maximum FRU Information Device size of 2^8 or 256 bytes).</td></tr>
</table>

8.3.2 Controller Level Reset

…

The actions performed on a Controller Level Reset are outlined in the NVM Express Base Specification. A Controller Level Reset shall have 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 shall not stop t~~The servicing of any ~~the~~ Management Interface Command Set commands, NVM Express Admin Command Set commands, or Control Primitives shall be independent of and not affected by any one or more Controllers in the NVM Subsystem being disabled or being reset by a Controller Level Reset unless the Management Endpoint servicing the NVMe-MI Request is reset (e.g., due to an NVM Subsystem Reset or due to a PCIe Reset of the PCIe VDM Management Endpoint servicing the NVMe-MI Request)~~ (e.g., NVM Express Admin Command Set commands are still serviced even though the NVMe Controller may be disabled or held in reset)~~. A Controller Level Reset shall have no effect on the AEM servicing model (refer to section 4.4).

A Controller Level Reset may prevent PCIe Commands from being processed on that Controller (refer to section 8.1). If a PCIe Command is prevented from being processed due to a Controller Level Reset that does not reset the Management Endpoint, then that PCIe Command shall be completed with status PCIe Inaccessible.

…

Description of Specification Changes for NVM Express Base Specification 2.0d

5.12 Firmware Commit command

…

Figure 182: Firmware Commit – Command Dword 10

<table>
<tr><th>Bits</th><th>Description</th></tr>
<tr><td>31</td><td><b>Boot Partition ID (BPID):</b> Specifies the Boot Partition that shall be used for the Commit Action, if applicable.</td></tr>
<tr><td>30:06</td><td>Reserved</td></tr>
<tr><td>05:03</td><td><b>Commit Action (CA):</b> This field specifies the action that is taken (refer to section 3.11) on the image downloaded with the Firmware Image Download command or on a previously downloaded and placed image. The actions are indicated in the following table.
<table>
<tr><th>Value</th><th>Definition</th></tr>
<tr><td>000b</td><td>Downloaded image replaces the existing image, if any, in the specified Firmware Slot. The newly placed image is not activated.</td></tr>
<tr><td>001b</td><td>Downloaded image replaces the existing image, if any, in the specified Firmware Slot. The newly placed image is activated at the next Controller Level Reset.</td></tr>
<tr><td>010b</td><td>The existing image in the specified Firmware Slot is activated at the next Controller Level Reset.</td></tr>
<tr><td>011b</td><td>Downloaded image replaces the existing image, if any, in the specified Firmware Slot and is then activated immediately. If there is not a newly downloaded image, then the existing image in the specified firmware slot is activated immediately. The Firmware Commit command remains in progress until image activation has completed successfully or unsuccessfully (i.e., the Firmware Commit command is not a background operation).</td></tr>
<tr><td>100b to 101b</td><td>Reserved</td></tr>
<tr><td>110b</td><td>Downloaded image replaces the Boot Partition specified by the Boot Partition ID field.</td></tr>
<tr><td>111b</td><td>Mark the Boot Partition specified in the BPID field as active and update BPINFO.ABPID.</td></tr>
</table></td></tr>
<tr><td>02:00</td><td><b>Firmware Slot (FS):</b> Specifies the firmware slot that shall be used for the Commit Action, if applicable. If the value specified is 0h, then the controller shall choose the firmware slot (i.e., slot 1 to slot 7) to use for the operation.</td></tr>
</table>

…

5.17.2.1 Identify Controller Data Structure (CNS 01h)

The Identify Controller data structure (refer to Figure 276) is returned to the host for the controller processing the command.

Figure 276: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O[1]	Admin[1]	Disc[1]	Description
				Controller Capabilities and Features
…	…	…	…	…

Figure 276: Identify – Identify Controller Data Structure, I/O Command Set Independent

Bytes	I/O[1]	Admin[1]	Disc[1]	Description
271:270	O	O	R	**Maximum Time for Firmware Activation (MTFA):** This field ~~I~~indicates the maximum time the controller temporarily stops processing commands to activate the firmware image. This field shall be valid if the controller supports firmware activation without a reset. This field is specified in 100 millisecond units. A value of 0h indicates that the maximum time is undefined. For the amount of time to process a Firmware Commit command that specifies a value of 011b in the Commit Action field (i.e., firmware activation without reset), refer to the Maximum Processing Time for Firmware Activation Without Reset field.
…	…	…	…	…
357:356	O	O	O	**Domain Identifier:** This field indicates the identifier of the domain (refer to section 3.2.4) that contains this controller. If the MDS bit is set to '1', then this field shall be set to a non-zero value. If the NVM subsystem does not support multiple domains (i.e., the NVM subsystem consists of a single domain), then this field shall be cleared to 0h.
361: 360	O	O	R	**Maximum Processing Time for Firmware Activation Without Reset (MPTFAWR):** This field shall indicate the estimated maximum time in 100 ms units required by the controller to process a Firmware Commit command that specifies a value of 011b in the Commit Action field (i.e., firmware activation without reset). This field applies to Firmware Commit commands received on an NVM Express controller Admin Submission Queue or received out-of-band on a Management Endpoint (refer to the NVM Express Management Interface Specification). If firmware activation without reset is not supported, then this field shall be cleared to 0h. A value of 0h indicates that no time is indicated. This field shall not be cleared to 0h on implementations that support firmware activation without reset and that are compliant with revisions later than 2.0 of this specification. <Note to Editor: Change "revisions later than 2.0" in the prior sentence to "revision 2.1 or later" in the integration specification.>
367:~~358~~ 362				Reserved
…	…	…	…	…