LEGAL NOTICE:

This NVM Express Management Interface 1.0a 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 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 - 2018 NVM Express, Inc. ALL RIGHTS RESERVED."** When making any such citations or references to this document you are not permitted to revise, alter, modify, make any derivatives of, or otherwise amend the referenced portion of this document in any way without the prior express written permission of NVM Express, Inc. Nothing contained in this document shall be deemed as granting you any kind of license to implement or use this document or the specification described therein, or any of its contents, either expressly or impliedly, or to any intellectual property owned or controlled by NVM Express, Inc., including, without limitation, any trademarks of NVM Express, Inc.

LEGAL DISCLAIMER:

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

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

NVM Express Workgroup
c/o Virtual, Inc.
401 Edgewater Place, Suite 600
Wakefield, MA 01880
info@nvmexpress.org

NVM Express™ Technical Errata

Errata ID	**002**
Revision Date	**4/8/2018**
Affected Spec Ver.	**NVM Express™ MI 1.0a & TP 6001**
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Austin Bolen	Dell EMC
Michael Allison	SK Hynix
Peter Onufryk	Microsemi Corporation

Errata Overview

- Fixed grammar in section 1.4
- Fixed grammar in Control primitives
- Added clarification in reference to servicing state usage.
- Replaced the undefined term “NVMe Data” and “NVMe Response/Request Data”
- Replaced the undefined term “input data”
- Updated Byte 0 reference in diagrams to align with SES TP 6001
- Added byte mapping clarifications to Section 1.5 of SES TP 6001

Revision History

Revision Date	Change Description
7/18/2017	• Fixed typo in section 1.4
7/30/2017	• Fixed grammar in Control primitives
8/30/2017	• Added clarity around the general work “state”. Removed servicing state usage before definition.
9/21/2017	• Replaced the undefined term “NVMe Data” • Replaced the term “input data”
9/25/2017	• Added a “<” to the Byte 0 reference in the diagrams to align with SES TP.
3/11/2018	• Added byte mapping clarification to Section 1.5 of TP 6001
3/18/2018	• Updated errata overview based on reviewer feedback
3/30/2018	• Changed all instances of “servicing state” to “command servicing state.” • Removed elimination of “and command servicing transitions back to the Idle state” text and added reference to Section 4.3 to the text in Section 4.4.
4/8/2018	• Removed section name in cross reference based on workgroup feedback.

Description of Specification Changes

Modify Section 1.4 (Architectural Model) as shown below:

Each Management Endpoint~~s~~ advertises its unique capabilities.

Modify portions of Figure 17 in Section 4.2 as shown below:

05h	**Invalid Command Size:** The Command Message body was larger or smaller than that expected by the command due to a reason other than too much or too little Request Data~~input data~~ (e.g., the command did not contain all the required parameters or no ~~input data~~Request Data was expected but the command message body is larger than that needed to contain the required parameters). The expected command message body size is determined by the command opcode assuming no other errors are detected (e.g., Invalid Command Opcode or Invalid Parameter).	Refer to 4.2.1
06h	**Invalid Command Input Data Size:** The Command Message requires Request Data~~input data~~ and contains too much or too little ~~input~~ data.	Refer to 4.2.1

Modify Section 4.4 as shown below:

A Management Controller sends a Command Message to a Management Endpoint that targets a specific Command Slot in the Management Endpoint. The Management Endpoint assembles MCTP packets into Command Messages targeting a Command Slot. The Command Slot remains allocated to the Command Message until servicing of the Command Message has completed and command servicing transitions back to the Idle state (refer to Section 4.3).

Modify Section 4.4 as shown below:

Control Primitives are Request Messages sent from a Management Controller to a Management Endpoint to affect the command processing flow. Control Primitives may target a Command Slot. Unlike Command Messages, Control Primitives may be sent while the Command Slot is in any command servicing state and are processed immediately by the Management Endpoint. Unless otherwise indicated, Control Primitives do not change the command servicing state of the Command Slot.

Modify Section 4.4.1 as shown below:

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:

Modify Section 4.4.1 as shown below:

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:

Modify Section 4.4.2 as shown below:

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

…

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:

Modify Section 4.4.3 as shown below:

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

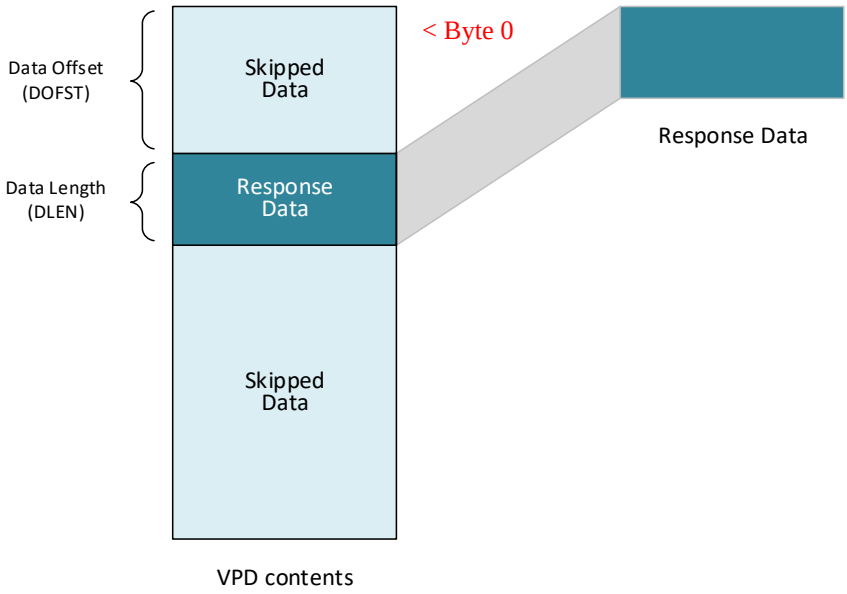
Modify Section 4.4.5 as shown below:

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

Modify a portion of Figure 72 in Section 5.7 as shown below:

<Editor Note: The author did not have the original drawings and used Word text box overlays to make the visible changes>

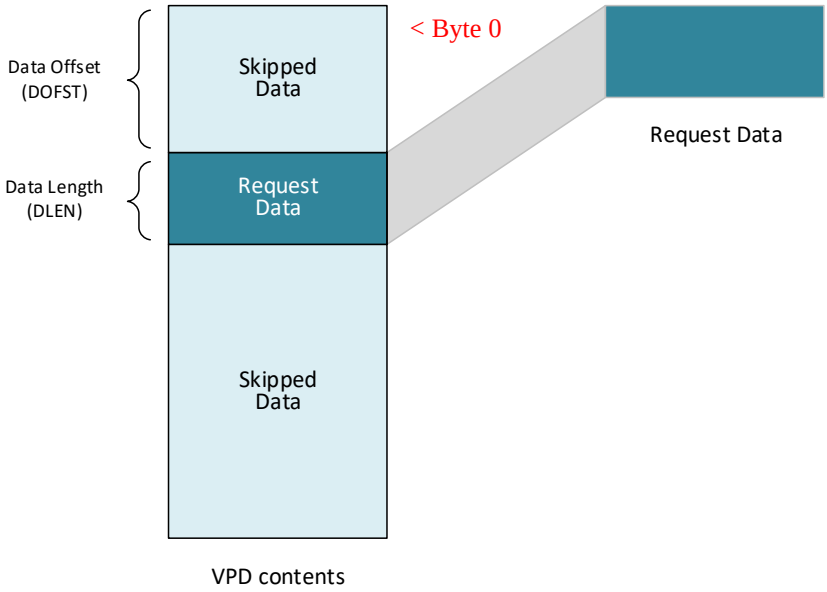
Figure 1: VPD Read Response Data



Modify a portion of Figure 75 in Section 5.8 as shown below:

<Editor Note: The author did not have the original drawings and used Word text box overlays to make the visible changes>

Figure 2: VPD Write Request Data



Modify portions of Figure 78 in Section 6 as shown below:

31:28	**Data Offset (DOFST):** For commands that transmit data from the Management Controller to the Management Endpoint (i.e., the RequestNVMe Data field in the Request Message has non-zero length) or do not transmit data, this field shall be cleared to '0'. If this field is not 0h, then the Management Endpoint shall return an error response with status Invalid Parameter. For commands that transmit data from the Management Endpoint to the Management Controller (i.e., the ResponseNVMe Data field in the Response Message has non-zero length), this field specifies the starting offset, in bytes, into the completion data contained in the Response Message. Bits 0 and 1 of this field shall be cleared to '0'.
35:32	**Data Length (DLEN):** For commands that do not transmit data in neither the Request Message nor Response Message, this field shall be cleared to 0h. If this field is not 0h, then the Management Endpoint shall return an error response with status Invalid Parameter. For commands that transmit data from the Management Controller to the Management Endpoint (i.e., the RequestNVMe Data field in the Request Message has non-zero length), this field specifies the length, in bytes, of the data contained in the Request Message. For commands that transmit data from the Management Endpoint to the Management Controller (i.e., the ResonseNVMe Data field in the Response Message has non-zero length), this field specifies the length, in bytes, of the data contained in the Response Message. Bits 0 and 1 of this field shall be cleared to '0'. This field shall be less than or equal to 4096.

Modify portions of Section 6.1 as shown below:

NVMe Admin Commands may contain data as part of the Command Message. This data is passed in the RequestNVMe Data field instead of using PRP Lists or SGL segments. The PRP Entry 2 (PRP2) and Metadata Pointer (MPTR) fields within the NVMe Admin Commands are reserved.

If there is no data sent with the NVMe Admin Command (e.g., the Data Transfer subfield for the opcode is 00b), then the Data Offset and Data Length fields shall be cleared to 0h.

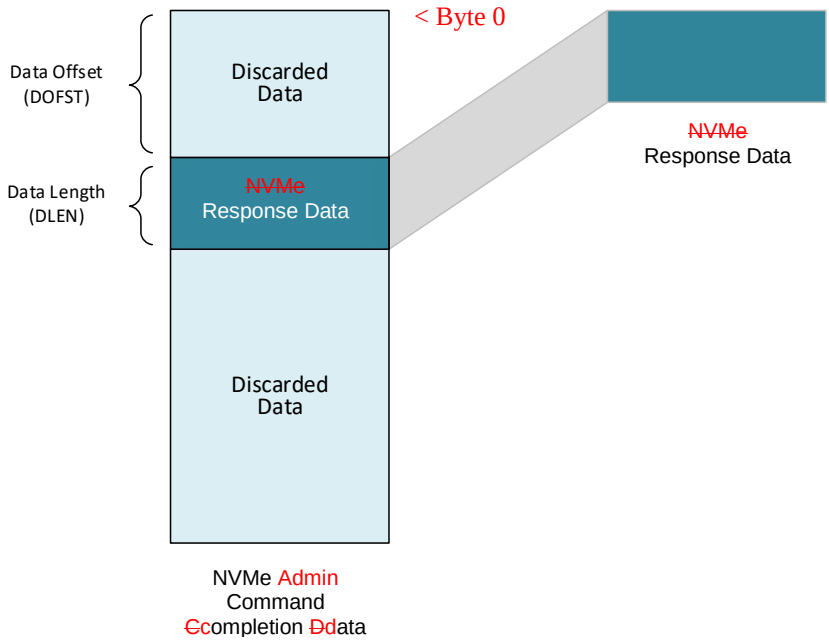
If there is data sent with the NVMe Admin Command (i.e., the Data Transfer subfield for the opcode is 01b), then the Data Offset field shall be 0h and the Data Length field shall be set to the length of the Request Datainput data required by the command. If the Data Length field does not correspond to the required length, the Management Endpoint shall respond with an Invalid Parameter error status response.

If there is Response Ddata expected in the Response Message in the completion of the NVMe Admin Command (i.e., the Data Transfer subfield in the corresponding NVMe Admin Command for the opcode is 10b), then the Data Offset and Data Length fields describe the portion of the NVMe Admin Command completion data that is transferred in the Response Message. Any remaining data not transferred in the Response Message is discarded by the Management Endpoint as shown in Figure 3. If the Data Length plus Data Offset fields are greater than the size of the NVMe Admin cCommand completion data, the Management Endpoint should respond with an Invalid Parameter error status response.

Modify a portions of Figure 81 in Section 6.1 as shown below:

<Editor Note: The author did not have the original drawings and used Word text box overlays to make the visible changes>

Figure 3: NVMe Admin Command Response Data Example



Modify portions of Section 6.1 as shown below:

- Invalid Command Input Data Size (e.g., the ~~NVMe~~ Request Data field is larger than the size specified in the Data Length field)

Modify portions of Section 1.5 in TP 6001 as shown below:

Refer to SES-3 for a list and description of SES control type diagnostic pages and SES status type diagnostic pages. The mapping of bytes in SES pages to NVMe-MI Request and Response Data is one-to-one where byte x of the SES page maps to byte x in the NVMe-MI Request or Response Data (e.g., byte zero of the SES control type diagnostic page corresponds to byte zero of NVMe-MI Request Data). The NVMe firmware update process is used (i.e., Firmware Image Download and Firmware Commit commands) to update NVMe firmware. Download Microcode Control and Status diagnostic pages, if supported, shall only be supported on Enclosure elements.