LEGAL NOTICE:

This NVM Express revision 1.2 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 NVM Express revision 1.2 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 - 2016 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	**001**
Revision Date	**1/25/2016**
Affected Spec Ver.	**NVM Express™ MI 1.0**
Corrected Spec Ver.	

Errata Author(s)

Name	Company
Austin Bolen	Dell
Peter Onufryk	PMC
John Carroll	Intel
Jason Gao	HGST
Myron Loewen	Intel
Judy Brock	Samsung
Chris Sneed	HGST

Errata Overview

This ECN addresses typos and requests for clarifications received during the 45-day NVMe-MI 1.0 member review period. Please refer to revision history for a detailed list of changes.

Revision History

Revision Date	Change Description
10/01/2015	Initial Errata
10/08/2015	Clarified section 1.4
10/22/2015	Clarified Invalid opcode
10/29/2015	Fixed Invalid Parameter typo. Added clarifications for SMBus addresses and commands to a non-idle command slot.
11/5/2015	Changed Command Processing language based on feedback from the workgroup. Edited Figure 30 to add Command Slot specific column
11/12/2015	Clarified Command Slot specific in Figure 30. Propagated reserved to Figure 75 and 76. Clarified Invalid Opcode edit
11/17/2015	Further edits to section 4.3
12/10/2015	Fixed figure 75 (removing SQE DW8)
1/13/2016	Added errata overview description

Description of Specification Changes

Modify a portion of Figure 91 (PCIe I/O Write) as shown below:

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:19</td><td>Reserved</td></tr>
<tr><td>18:16</td><td>Base Address Register (BAR): 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 Configuraiton 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 lower 32-bits of the BAR.
<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-7h</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>15:00</td><td>Length (LENGTH): This field specifies the number of bytes to be ~~read~~ written.</td></tr>
</table>

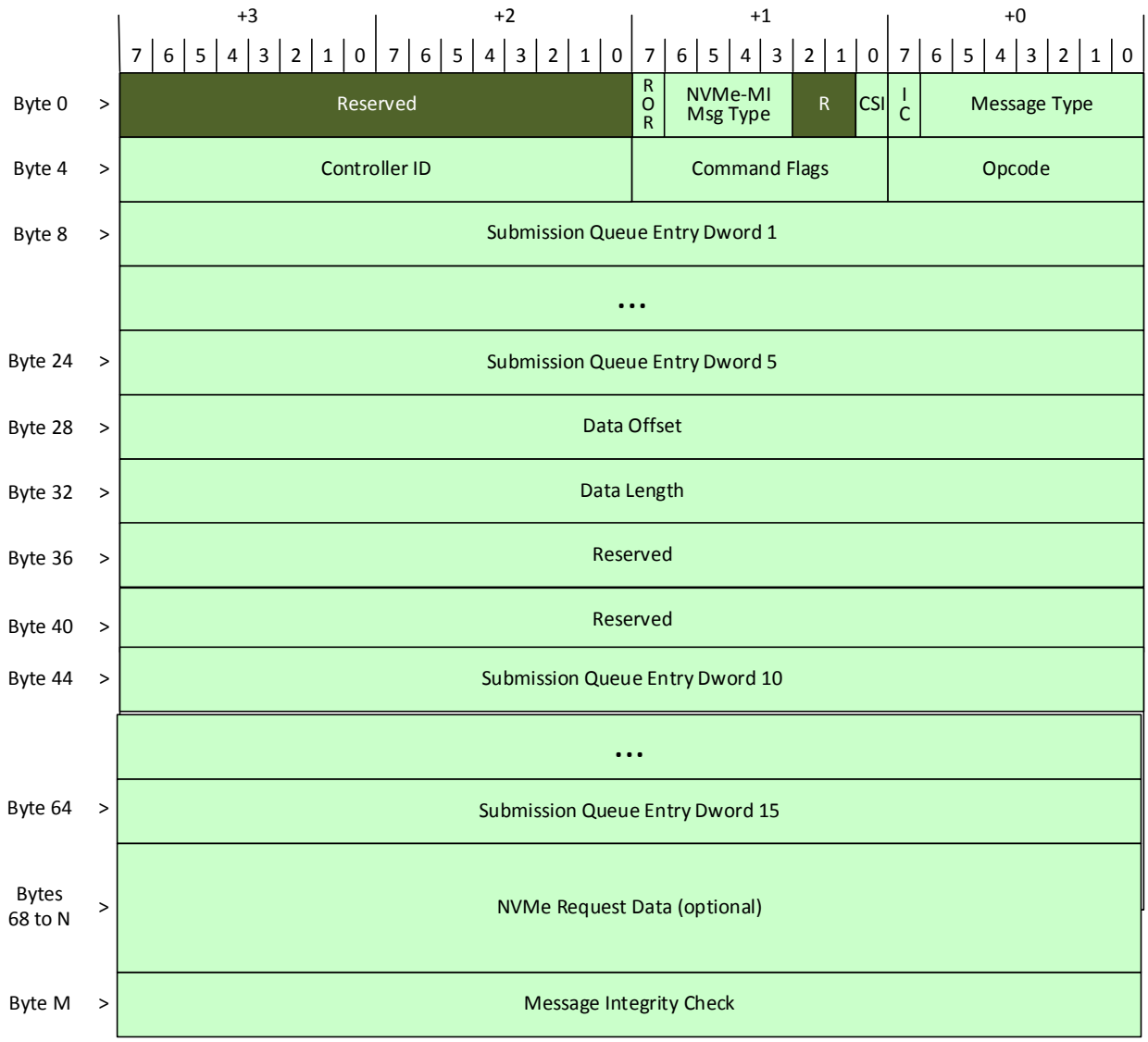
Modify a portion of Figure 96 (PCIe Memory Write) as shown below:

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:19</td><td>Reserved</td></tr>
<tr><td>18:16</td><td>Base Address Register (BAR): This field specifies the PCI Base Address Register (BAR) of the memory space to be written. BARs are located beginning at 10h in PCI Configuraiton 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 lower 32-bits of the BAR.
<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-7h</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>15:00</td><td>Length (LENGTH): This field specifies the number of bytes to be ~~read~~ written.</td></tr>
</table>

Modify a portion of section 6.1 (NVMe Admin Command Request and Response Data) as shown below:

NVMe Admin Commands may contain data as part of the Command Message. This data is passed in the NVMe 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.

Replace Figure 75 (NVMe Admin Command Request Format) as shown below:



Modify a portion of Figure 76 (NVMe Admin Command Request Description) as shown below:

Byte	Description
~~39:36~~	~~**Submission Queue Entry Dword 8 (SQEDW8):** Submission Queue Entry Dword 8 as defined in the NVMe specification~~
43:~~40~~ 36	~~**Submission Queue Entry Dword 9 (SQEDW9):** Submission Queue Entry Dword 9 as defined in the NVMe specification~~ Reserved
47:44	**Submission Queue Entry Dword 10 (SQEDW10):** Submission Queue Entry Dword 10 as defined in the NVMe specification

Modify a portion of section 1.4 (Architectural Model) as shown below:

An NVMe storage device, such as a PCIe SSD, that implements this specification, consists of an ~~NVMe~~ NVM Subsystem with one or more ~~PCIe~~ Management Endpoints ~~ports~~. There may be up to one Management Endpoint per PCIe port and ~~and~~ ~~an~~ ~~optional~~ SMBus/I2C port. Each Management Endpoint ~~port~~ has a Port Identifier that is less than or equal to the Number of Ports (NUMP) field value in the NVM Subsystem Information Data Structure. The ~~p~~Port ~~i~~Identifier for a PCIe port is the same as the Port Number field in the PCIe Link Capabilities Register.

Modify a portion of Figure 17 (Response Message Status Values) as shown below:

Value	Description	Error Reponse Format
03h	**Invalid Command Opcode:** ~~Invalid command opcode field value.~~ The associated command opcode field is not valid. Invalid opcodes include reserved and optional opcodes that are not implemented.	Refer to **Error! Reference source not found.**

Modify a portion of section 4.2.2 (Invalid Parameter Error Response) as shown below:

An invalid parameter error response is generated for error responses where the Status field is set to ~~03h (i.e.,~~ Invalid Parameter~~)~~.

Modify a portion of section 1.5 (Conventions) as shown below:

Some fields or registers are 0's based values. In a 0's based value, the value of 0h corresponds to 1; other values similarly correspond to the value+1.

SMBus/I2C addresses are written as 8-bit hex values where bits 7:1 contain the 7-bit SMBus/I2C address and bit 0 is cleared to 0b.

Modify a portion of section 2.2 (SMBus/I2C) as shown below:

The SMBus/I2C Management Endpoint shall be accessible at a power-up SMBus/I2C address of ~~0x~~3Ah and should be SMBus ARP-capable (as defined in the SMBus 3.0 specification). If the NVM Subsystem is "Discoverable" (as defined in the SMBus 3.0 specification), the device shall issue a "Notify ARP Master" command when the NVM Subsystem is ready to communicate.

If the NVM Subsystem implements an SMBus/I2C interface, then VPD information shall be accessible from the Management Endpoint using Sequential Read and Random Read operations as defined by the IPMI Platform Management FRU Information Storage Definition specification.

The VPD shall be accessible using I2C read operations from a FRU Information Device at a power-up SMBus/I2C address of ~~0x~~A6h and should be SMBus ARP-capable (as defined in the SMBus 3.0 specification). If the FRU Information Device is "Discoverable" (as defined in the SMBus 3.0 specification), it shall issue a "Notify ARP Master" command when the FRU Information Device is ready to communicate.

Modify a portion of section 4.3 (Command Processing Model) as shown below:

NVMe-MI utilizes Command Slots for command servicing ~~processing~~. ~~Command Slots are logically used for MCTP NVMe-MI Request Message and Response Message assembly. Together with the request/response processing model, Command Slots provide a mechanism for message flow control.~~ A Management Controller should not send a new Command Message to a Command Slot until the Response Message for the previously issued command to that Command Slot has been received. ~~Associated with each~~ Each Management Endpoint ~~are~~ contains 2 Command Slots~~. Each Command Slot~~ that each include~~s a~~ state information and a Pause flag (refer to 4.4.4).

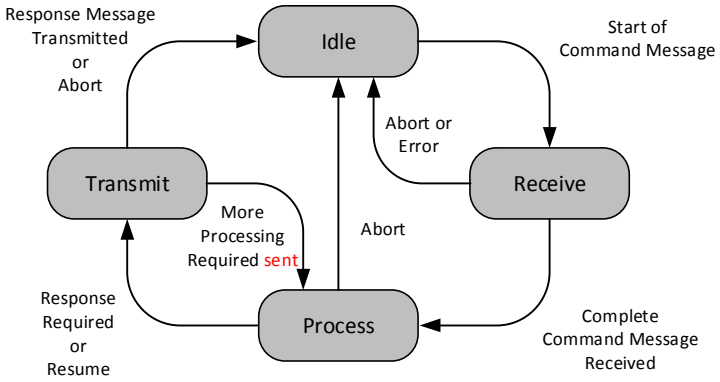
A Management Controller sends a ~~Request~~ Command Message to a Management Endpoint ~~and~~ that targets a specific Command Slot in the Management Endpoint. The Management Endpoint assembles MCTP packets into Command Messages targeting a Command Slot. ~~When a Management Endpoint receives MCTP packets for a Command Message that target a Command Slot, the packets are kept in a buffer associated with that Command Slot to be assembled.~~ The Command Slot remains allocated to the Command Message until

~~processing~~ servicing of the ~~command~~ Command Message has completed~~, the associated Response Message has been transmitted,~~ and ~~the Command Slot~~ command servicing transitions back to the Idle state.

A Command Message is the only type of multi-packet MCTP NVMe-MI message that may be received by a Management Endpoint. The maximum number of Command Messages in flight to a Management Endpoint is equal to the number of Command Slots. The operation of each Command Slot is independent, allowing a Management Controller to have 2 independent streams of Command Messages to a Management Endpoint. The Command Message associated with each Command Slot are processed in parallel. If the NVM Subsystem implements multiple Management Endpoints, then command ~~processing~~ servicing of each Management Endpoint~~s~~ occurs in parallel. A NVM Subsystem that implements *N* Management Endpoints may have up to 2*N* ~~commands executing~~ Command Messages serviced in parallel.

The Command Servicing State Diagram in Figure 21 is used to describe functional requirements and does not mandate an implementation. ~~A Command Slot may be in 1 of 4 possible states. These states as well as valid state transitions are shown in Figure 1.~~

Figure 1: Command Servicing ~~Slot~~ State Diagram



1. **Idle**: ~~The state when there is no Command Message associated with the Command Slot.~~ This is the default state of the command servicing state machine ~~a Command Slot~~ (e.g., following a reset). ~~A~~ Command servicing ~~Slot~~ transitions from Idle to the Receive state when the first MCTP packet of a MCTP NVMe-MI command message is received (i.e., an MCTP packet with the SOM bit in the MCTP packet header set to '1'~~,~~ and the Message Type set to 4h~~, and the CSI field set to the corresponding Command Slot Identifier~~).

2. **Receive**: The state when the first packet of a Command Message has been received and the message is being assembled and/or validated. ~~A~~ Command servicing ~~Slot~~ transitions from Receive to the Idle state when an Abort Control Primitive is received, an error is detected in message assembly (refer to 3.2.2), or the Message Integrity Check fails (refer to 3.2.1.1). ~~A~~ Command servicing ~~Slot~~ transitions from Receive state to the Process state when a Command Message is assembled and the message integrity check is successful.

3. **Process**: The state when a Command Message is processed. Processing of a command consists of performing the actions specified by the command or aborting the command. ~~A~~ Command servicing ~~Slot~~ transitions from Process to the Transmit state when a response is required (i.e., the Pause Flag is cleared to '0' and either of the following are true: all processing of the command has completed or command processing is expected to exceed the corresponding transport binding specification response timeout). ~~A~~ Command servicing ~~Slot~~ transitions from the Process state to the Idle state due to an Abort Control Primitive (refer to 4.4.3).

4. **Transmit**: The state in which a Response Message for the Command Message is transmitted to the Management Controller. ~~A~~ Command servicing ~~Slot~~ transitions from the Transmit to the Idle state once the entire MCTP message associated with the response to the command has been transmitted on the physical medium or due to an Abort Control Primitive (refer to 4.4.3). If command ~~processing~~ servicing did not complete in the Process state, then the Management Endpoint transmits a response with status More Processing Required and ~~the Command Slot~~ command servicing transitions back to the Process state.

~~Receiving a new Command Message "start" packet (packet with SOM = 1b) to the same Command Slot while a Command Message is being assembled (i.e., in the Receive state) terminates the original message assembly. All data for the terminated Command Message is discarded. The newly received start packet is not dropped, but instead it begins a new message assembly. This~~ The behavior of receiving two or more overlapping Command Messages to the same Command Slot is undefined. If this results in the Management Endpoint discarding a Command Message, then this is considered receiving a Command Message to a non-Idle Command Slot (CMNICS). Refer to section 4.4.4.

~~If a Command Message packet is received when the corresponding Command Slot is in the Process or Transmit state, then the Management Endpoint discards the Command Message packet without a response. This is also considered receiving a Command Message to a non-Idle Command Slot (CMNICS). Refer to section 4.4.4.~~

Modify a portion of section 4.4.4 (Get State) as shown below:

~~Bits 04 through 13 are global for the Management Endpoint and indicate MCTP transport errors that have occurred. Refer to the MCTP Base Specification section for Dropped Packets and Dropped Messages for details on the errors.~~

Figure 2: Get State Control Primitive Success Response Message Fields

Bits	CS Specific[1]	Description
15	Yes	**Pause Flag (PFLG):** This field indicates whether or not the Command Slot is paused. A '1' in this field indicates the Command Slot is paused. A '0' in this field 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	**NVM Subsystem Reset Occurred (NSSRO):** This field indicates when an NVM Subsystem Reset occurs while main power is applied. This field is set to '1' if the last occurrence of an NVM Subsystem Reset occurred while main power was applied to the NVM Subsystem. This field is cleared to '0' following a power cycle and following a Get State primitive with the CESF field set to '1'.
13	No	**Bad Packet or Other Physical Layer (BPOPL):** This field 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 executed with the CESF field set to '1'.
12	No	**Bad, Unexpected, or Expired Message Tag (BUEMT):** This field 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 executed with the CESF field set to '1'.
11	No	**Out-of-Sequence Packet Sequence Number (OSPSN):** This field 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 executed with the CESF field set to '1'.
10	No	**Unexpected Middle or End of Packet (UMEP):** This field 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 executed with the CESF field set to '1'.
09	No	**Incorrect Transmission Unit (ITU):** This field 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 executed with the CESF field set to '1'.
08	No	**Unknown Destination ID (UDSTID):** This field 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 executed with the CESF field set to '1'.
07	No	**Bad Header Version (BHVS):** This field 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 executed with the CESF field set to '1'.
06	No	**Unsupported Transmission Unit (UTUNT):** This field 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 executed with the CESF field set to '1'.
05	No	**Timeout Waiting for a Packet (WPTT):** This field 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 executed with the CESF field set to '1'.
04	No	**Bad Message Integrity Check Error (~~TMICE~~ BMICE):** This field 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 executed with the CESF field set to '1'.
03	No	**Command Message to non-Idle Command Slot (CMNICS):** This field is set to '1' if the Management Endpoint ~~received~~ discarded one or more Command Messages due to overlapping ~~a~~ Command Message~~s packet while the~~ to a Command Slot ~~is not in the Idle state~~ since the last time Get State primitive was executed with the CESF field set to '1'.
02		Reserved
01:00	Yes	**Slot State (SSTA):** This field indicates the current state of the Command Slot. An implementation may choose to indicate only the Idle and Process states in this field. **Value** / **Description** 0h – Idle 1h – Receive 2h – Process 3h – Transmit

Notes:
1. Command Slot Specific. Yes in this column indicates the value of the field within a Management Endpoint is independent per Command Slot.