NVMe Management Interface (NVMe-MI)

Peter Onufryk
Microsemi Corp.
NVMe-MI Workgroup Chair

Austin Bolen
Dell EMC
NVMe-MI Workgroup Vice Chair
NVM Express, Inc.
120+ Companies defining NVMe together

Board of Directors
13 elected companies, stewards of the technology & driving processes
Chair: Amber Huffman

Technical Workgroup
NVMe Base and NVMe Over Fabrics
Chair: Amber Huffman

Management Intf. Workgroup
Out-of-band management over SMBus and PCIe® VDM
Chair: Peter Onufryk
Vice Chair: Austin Bolen

Marketing Workgroup
NVMexpress.org, webcasts, tradeshows, social media, and press
Co-Chairs: Janene Ellefson and Jonmichael Hands

Interop (ICC) Workgroup
Interop & Conformance Testing in collaboration with UNH-IOL
Chair: Ryan Holmqvist
RASM (Reliability, Availability, Serviceability, Manageability)

“Customers choose suppliers who provide the features that are important to them. Customers care about TCO (Total Cost of Ownership). Consequently, in the server space, MHz is not the only thing that’s important: TCO is greatly affected by the RASM features of the servers. When server OEMs and users talk, their focus is RASM: Reliability, Availability, Serviceability, and Manageability. To a customer, RASM means dollars. Adding or improving on RASM reduces TCO.

The cost of downtime is extremely high. According to IMEX Research*, the average cost of an unplanned outage runs into the hundreds of thousand of dollars.” (Reference 2)
Management Fundamentals

Pillars of Systems Management

• Inventory
• Configuration
• Monitoring
• Change Management

Management Operational Times

• Deployment (No OS)
• Pre-OS (e.g. UEFI/BIOS)
• Runtime
• Auxiliary Power
• Decommissioning
What is the NVMe Management Interface 1.0a?

A programmable interface that allows out-of-band management of an NVMe Storage Device Field Replaceable Unit
NVM Express Roadmap

**NVM Express (NVMe)**
- **2014**
  - Q1: NVMe 1.2 Nov’14
    - Namespace Management
    - Controller Memory Buffer
    - Host Memory Buffer
    - Live Firmware Update

- **2015**
  - Q2: NVMe 1.2.1 May’16
    - Sanitize
    - Streams
    - Virtualization

- **2016**
  - Q3: NVMe-oF 1.0 May’16
    - Transport and protocol
    - RDMA binding

- **2017**
  - Q4: NVMe 1.3 May’17
    - IO Determinism
    - Persistent Memory Region
    - Persistent Event Log
    - Multipathing

- **2018**
  - Q4: NVMe (next)
    - Enhanced Discovery
    - In-band Authentication
    - TCP Transport Binding

- **2019**
  - Q4: NVMe-oF (next)

**NVMe Mgmt. Interface (NVMe-MI)**
- **2014**
  - Q1: NVMe-MI 1.0 Nov’15
    - Out-of-band management
    - Device discovery
    - Health & temp monitoring
    - Firmware Update

- **2016**
  - Q4: NVMe-MI 1.0a April’17
    - Enclosure Management
    - In-band Mechanism
    - Storage Device Extension

- **2017**
  - Q4: NVMe-MI 1.1

**Languages**
- Subject to change

- Released NVMe specification
- Planned release
Benefits of NVMe-MI and Standardization

<table>
<thead>
<tr>
<th>Benefit</th>
<th>OEM</th>
<th>Drive Vendor</th>
<th>End User</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear requirements and specification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Industry standard compliance program</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Industry standard tools</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ability to source NVMe-MI drives from multiple vendors</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reduces need for drive vendors to develop proprietary</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>management features</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lower TCO over life of NVMe Storage Device</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Allows product differentiation</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
# Sample End User Use Cases

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inventorying</td>
<td>Asset management. Re-provisioning systems. Track quality of components.</td>
</tr>
<tr>
<td>Health monitoring</td>
<td>Identify bad drives for quick replacement.</td>
</tr>
<tr>
<td>Wear monitoring</td>
<td>Replace drives nearing wear-out to avoid failure.</td>
</tr>
<tr>
<td>Temp. monitoring</td>
<td>Fan throttling reduces power, noise, and fan wear.</td>
</tr>
<tr>
<td>Power monitoring and configuration</td>
<td>Power throttling to save energy and cool system.</td>
</tr>
<tr>
<td>Perf. monitoring</td>
<td>Look for performance bottlenecks.</td>
</tr>
<tr>
<td>Configuring</td>
<td>Format drives for initial use. Crypto erase drives for re-provisioning or decommissioning.</td>
</tr>
<tr>
<td>Change Mgmt.</td>
<td>Update drive firmware for bug fixes and security patches.</td>
</tr>
</tbody>
</table>
Field Replaceable Unit (FRU)

FRU Definition (Wikipedia)

A circuit board, part or assembly that can be quickly and easily removed from a computer or other piece of electronic equipment, and replaced by the user or a technician without having to send the entire product or system to a repair facility.
NVM Subsystem - one or more controllers, one or more namespaces, one or more PCI Express ports, a non-volatile memory storage medium, and an interface between the controller(s) and non-volatile memory storage medium.
NVMe Storage Device

- **NVM Storage Device** – One NVM Subsystem with one or more ports and an optional SMBus/I2C interface

Single Ported PCIe SSD

Dual Ported PCIe SSD with SMBus/I2C
Vital Product Data (VPD)

- Utilizes IPMI Platform Management FRU Information Storage Definition with NVMe-MI extensions

- The VPD may be accessed using two methods
  - NVMe-MI commands over MCTP
  - SMBus/I2C interface using I2C operations as defined by IMPI Platform Management FRU Information Storage Definition

<table>
<thead>
<tr>
<th>VPD Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Common Header</td>
</tr>
<tr>
<td>Product Info Area (optional)</td>
</tr>
<tr>
<td>NVMe MultiRecord Area</td>
</tr>
<tr>
<td>NVMe PCIe Port MultiRecord Area</td>
</tr>
<tr>
<td>Internal Use Area (optional)</td>
</tr>
<tr>
<td>Chassis Info Area (optional)</td>
</tr>
<tr>
<td>Board Info Area (optional)</td>
</tr>
</tbody>
</table>
Out-of-Band Management and NVMe-MI

- **Out-of-Band Management** – Management that operates with hardware resources and components that are *independent of the operation system control*

- **NVMe Out-of-Band Management Interfaces**
  - SMBus/I2C
  - PCIe Vendor Defined Messages (VDM)
  - IPMI FRU Data (VPD) accessed over SMBus/I2C
NVMe-MI Protocol Layering

Management Applications (e.g., Remote Console)

Management Controller (BMC)

NVMe Management Interface

Management Component Transport Protocol (MCTP)

MCTP over SMBus/I2C Binding

MCTP over PCIe VDM Binding

SMBus/I2C

PCIe

PCIe SSD

Application Layer

Protocol Layer

Transport Layer

Physical Layer
SMBus/I2C Topologies and Addressing

- **During Auxiliary Power (if supported)**
  - I2C serial EEPROM read/write access at default SMBus/I2C address 0xA6, but may be modified using ARP

- **During Main Power**
  - MCTP Endpoint at default SMBus/I2C address 0x3A, but may be modified using ARP
  - I2C serial EEPROM read/write access
    - If auxiliary power was provided, then SMBus/I2C address shall be maintained if modified using ARP; otherwise, the default address is 0xA6
    - SMBus/I2C address may be modified using ARP
NVMe-MI Message

<table>
<thead>
<tr>
<th>Byte 3</th>
<th>Byte 2</th>
<th>Byte 1</th>
<th>Byte 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>NVMe-MI Msg Type</td>
<td>R</td>
<td>CIC</td>
</tr>
</tbody>
</table>

Message Data

Message Integrity Check

NVMe-MI MCTP Message Assembly

1st MCTP Packet of Message

Physical Medium Specific Header

MCTP Packet Header

MCTP Packet Payload

Physical Medium Specific Trailer

2nd MCTP Packet of Message

Physical Medium Specific Header

MCTP Packet Header

MCTP Packet Payload

Physical Medium Specific Trailer

3rd MCTP Packet of Message

Physical Medium Specific Header

MCTP Packet Header

MCTP Packet Payload

Physical Medium Specific Trailer

4th MCTP Packet of Message

Physical Medium Specific Header

MCTP Packet Header

MCTP Packet Payload

Physical Medium Specific Trailer
NVMe-MI Message Taxonomy

- NVMe-MI Message
  - Request Message
    - Command Message
      - NVMe-MI Command
      - NVMe Admin Command
      - PCIe Command
  - Control Primitive
  - Success Response
  - Error Response
- Response Message
NVMe Storage Device

Single Ported PCIe SSD

Dual Ported PCIe SSD with SMBus/I2C
Command Servicing State Diagram for Command Slots

- **Idle**
  - Start of Command Message
  - Abort or Error

- **Transmit**
  - Response Message Transmitted or Abort
  - More Processing Required or Sent
  - Response Required or Resume

- **Process**
  - Abort
  - Complete Command Message Received

- **Receive**
## Control Primitives

<table>
<thead>
<tr>
<th>Control Primitive</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pause</td>
<td>Suspend transmission</td>
</tr>
<tr>
<td>Resume</td>
<td>Resume paused transmission</td>
</tr>
<tr>
<td>Abort</td>
<td>Reinitialize command slot</td>
</tr>
<tr>
<td>Get State</td>
<td>Retrieve state (e.g., errors) associated with a command slot</td>
</tr>
<tr>
<td>Replay</td>
<td>Retransmit response message for last command message processed in a command slot</td>
</tr>
</tbody>
</table>
## NVMe-MI 1.0a Command Set Overview

### Command Type: NVMe Management Interface Specific Commands

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read NVMe-MI Data Structure</td>
</tr>
<tr>
<td>NVM Subsystem Health Status Poll</td>
</tr>
<tr>
<td>Controller Health Status Poll</td>
</tr>
<tr>
<td>Configuration Get</td>
</tr>
<tr>
<td>Configuration Set</td>
</tr>
<tr>
<td>VPD Read</td>
</tr>
<tr>
<td>VPD Write</td>
</tr>
<tr>
<td>Reset</td>
</tr>
<tr>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>

### Command Type: PCIe Command

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe Configuration Read</td>
</tr>
<tr>
<td>PCIe Configuration write</td>
</tr>
<tr>
<td>PCIe I/O Read</td>
</tr>
<tr>
<td>PCIe I/O Write</td>
</tr>
<tr>
<td>PCIe Memory Read</td>
</tr>
<tr>
<td>PCIe Memory Write</td>
</tr>
<tr>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>

### Command Type: NVMe Admin Commands

<table>
<thead>
<tr>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Activate/Commit</td>
</tr>
<tr>
<td>Firmware Image Download</td>
</tr>
<tr>
<td>Format NVM</td>
</tr>
<tr>
<td>Get Features</td>
</tr>
<tr>
<td>Get Log Page</td>
</tr>
<tr>
<td>Identify</td>
</tr>
<tr>
<td>Namespace Management</td>
</tr>
<tr>
<td>Namespace Attachment</td>
</tr>
<tr>
<td>Security Send</td>
</tr>
<tr>
<td>Security Receive</td>
</tr>
<tr>
<td>Set Features</td>
</tr>
<tr>
<td>Vendor Specific</td>
</tr>
</tbody>
</table>
## NVMe Management Interface Specific Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>O/M*</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read NVMe-MI Data Structure</td>
<td>M</td>
<td>Retrieve information about the NVM Subsystem, Management Endpoint, or NVMe Controllers</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• NVM Subsystem Information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Port Information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Controller Information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Optional Commands Supported</td>
</tr>
<tr>
<td>NVM Subsystem Health Status Poll</td>
<td>M</td>
<td>Used to efficiently determine changes in health status attributes associated with the NVM Subsystem (e.g., Unrecoverable error, reset required, PCIe status, Controller SMART / Health Information, composite temperature, composite, and controller status)</td>
</tr>
<tr>
<td>Controller Health Status Poll</td>
<td>M</td>
<td>Efficiently determines changes in health status attributes associated with one or more Controllers in the NVM Subsystem</td>
</tr>
<tr>
<td>Configuration Get</td>
<td>M</td>
<td>Get NVMe-MI configuration parameter (e.g., SMBus/I2C frequency and MCTP transmission unit size)</td>
</tr>
<tr>
<td>Configuration Set</td>
<td>M</td>
<td>Set NVMe-MI configuration parameter</td>
</tr>
<tr>
<td>VPD Read</td>
<td>M</td>
<td>Read Vital Product Data (VPD)</td>
</tr>
<tr>
<td>VPD Write</td>
<td>M</td>
<td>Write Vital Product Data (VPD)</td>
</tr>
<tr>
<td>Reset</td>
<td>O</td>
<td>Reset NVM Subsystem</td>
</tr>
</tbody>
</table>

* O = Optional, M=Mandatory
# NVMe Admin Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>O/M</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Firmware Activate/Commit</td>
<td>O</td>
<td>Verifies that a valid firmware image has been downloaded and commits that revision to a specific firmware slot</td>
</tr>
<tr>
<td>Firmware Image Download</td>
<td>O</td>
<td>Download all of a portion of a firmware image for a future update to the controller</td>
</tr>
<tr>
<td>Format NVM</td>
<td>O</td>
<td>Low level format of the NVM media associated with one or more Namespaces</td>
</tr>
<tr>
<td>Get Features</td>
<td>M</td>
<td>Get NVMe configuration parameter</td>
</tr>
<tr>
<td>Set Features</td>
<td>O</td>
<td>Set NVMe configuration parameter</td>
</tr>
<tr>
<td>Get Log Page</td>
<td>M</td>
<td>Retrieve NVMe log page</td>
</tr>
<tr>
<td>Identify</td>
<td>M</td>
<td>Retrieve information about the Controllers, Namespaces, or NVM Subsystem</td>
</tr>
<tr>
<td>Namespace Management</td>
<td>O</td>
<td>Create or delete a Namespace</td>
</tr>
<tr>
<td>Namespace Attachment</td>
<td>O</td>
<td>Attach or detach a Namespace from a Controller</td>
</tr>
<tr>
<td>Security Send</td>
<td>O</td>
<td>Transfer command/data associated with security protocol</td>
</tr>
<tr>
<td>Security Receive</td>
<td>O</td>
<td>Transfer command/data associated with security protocol</td>
</tr>
</tbody>
</table>

*O = Optional, M=Mandatory*
## PCIe Commands

<table>
<thead>
<tr>
<th>Command</th>
<th>O/M</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCIe Configuration Read</td>
<td>O</td>
<td>Read PCI Express configuration space</td>
</tr>
<tr>
<td>PCIe Configuration Write</td>
<td>O</td>
<td>Write PCI Express configuration space</td>
</tr>
<tr>
<td>PCIe I/O Read</td>
<td>O</td>
<td>Read PCI Express I/O space</td>
</tr>
<tr>
<td>PCIe I/O Write</td>
<td>O</td>
<td>Write PCI Express I/O space</td>
</tr>
<tr>
<td>PCIe Memory Read</td>
<td>O</td>
<td>Read PCI Express memory space (BAR memory &amp; MMIO)</td>
</tr>
<tr>
<td>PCIe Memory Write</td>
<td>O</td>
<td>Write PCI Express memory space (BAR memory &amp; MMIO)</td>
</tr>
</tbody>
</table>

*O = Optional, M=Mandatory
## NVMe-MI Operational Times

### Power States

<table>
<thead>
<tr>
<th>Power State</th>
<th>Main Power</th>
<th>Auxiliary Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Powered Off</td>
<td>Off</td>
<td>Off</td>
</tr>
<tr>
<td>Auxiliary Power</td>
<td>Off</td>
<td>On</td>
</tr>
<tr>
<td>Main Power</td>
<td>On</td>
<td>On</td>
</tr>
<tr>
<td>Main Power with No Auxiliary Power</td>
<td>On</td>
<td>Off</td>
</tr>
</tbody>
</table>

### Operations Supported During Power States

<table>
<thead>
<tr>
<th>Operation</th>
<th>Powered Off</th>
<th>Auxiliary Power</th>
<th>Main Power (with Aux Power)</th>
<th>Main Power with No Aux Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>VPD I2C Access</td>
<td>Not Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Implementation Specific</td>
</tr>
<tr>
<td>SMBus/I2C MCTP Access</td>
<td>Not Supported</td>
<td>Optional(^1)</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>PCIe MCTP Access</td>
<td>Not Supported</td>
<td>Not Supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>

**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.
New Features Targeted for NVMe-MI 1.1

- In-Band NVMe-MI
- Enclosure Management
- NVMe Storage Device Enhancement
Out-of-Band Management and NVMe-MI

- **Out-of-Band Management** – Management that operates with hardware resources and components that are *independent of the operation system control*

- **NVMe Out-of-Band Management Interfaces**
  - SMBus/I2C
  - PCIe Vendor Defined Messages (VDM)
  - IPMI FRU Data (VPD) accessed over SMBus/I2C
In-Band Management and NVMe-MI

- In-band mechanism allows application to tunnel NVMe-MI commands through NVMe driver
  - Two new NVMe Admin commands
    - NVMe-MI Send
    - NVMe-MI Receive

- Benefits
  - Provides management capabilities not available in-band via NVMe commands
    - Efficient NVM Subsystem health status reporting
    - Ability to manage NVMe at a FRU level
    - Vital Product Data (VPD) access
    - Enclosure management
Example Enclosure
Enclosure Management

- Native PCIe Enclosure Management (NPEM)
  - Transport specific basic enclosure management
  - Submitted to the PCI-SIG Protocol Workgroup (PWG) on behalf of the NVMe Management Interface Workgroup
  - Approved by PCI-SIG on August 10, 2017

- SES Based Enclosure Management
  - Technical proposal being developed in NVMe-MI workgroup
  - Comprehensive enclosure management
SES Based Enclosure Management

- **SCSI Enclosure Services (SES)** is a standard developed by T10 for management of enclosures using the SCSI architecture.

- While the NVMe and SCSI architectures differ, the elements of an enclosure and the capabilities required to manage these elements are the same:
  - Example enclosure elements: power supplies, fans, display or indicators, locks, temperature sensors, current sensors, voltage sensors, and ports.

- **NVMe-MI** leverages SES for enclosure management:
  - SES manages the elements of an enclosure using control and status diagnostic pages transferred using SCSI commands (SCSI SEND DIAGNOSTIC & SCSI RECEIVE DIAGNOSTIC RESULTS).
  - NVMe-MI uses these same control and status diagnostic pages, but transfers them using the SES Send and SES Receive commands.
Enclosure Management Protocol Layering

- Legacy SCSI Host Software
- SCSI Translation
- PCIe / Fabric
- NVMe Controller with In-Band NVMe-MI Support
- NVMe-MI
- SES Send & SES Receive Commands
- SCSI Enclosure Services (SES)
- Management Controller
NVMe-MI 1.0a NVMe Storage Device

- **NVM Storage Device** – One NVM Subsystem with one or more ports and an optional SMBus/I2C interface
NVMe Storage Device with Multiple NVM Subsystems

M.2 Carrier Board from Amfeltec

ANA Carrier Board from Facebook
Multiple NVM Subsystems on an NVMe Storage Device and I2C/SMBus Topologies

Shared SMBus/ I2C

Segmented SMBus/ I2C
NVMe-MI Ecosystem

- Commercial test equipment and conformance tests exist for NVMe-MI
- NVMe-MI 1.0a compliance testing program has been developed
  - Compliance testing started in the May 2017 NVMe Plugfest conducted by the University of New Hampshire Interoperability Laboratory (UNH-IOL)
  - 6 devices have passed compliance testing and are on the NVMe-MI Integrators List
- Servers are shipping that support NVMe-MI
Summary

- NVMe-MI 1.0a has been released
  - Focused on managing NVMe Storage Devices (e.g., SSDs)
  - SSDs and systems are shipping that support NVMe-MI 1.0a

- NVMe-MI 1.1 is nearing completion
  - Technical work is scheduled for completion this year and a ratified specification is expected in Q1’18
  - Key new features in NVMe-MI 1.1
    - In-band NVMe-MI
    - Enclosure Management
    - NVMe Storage Device Enhancements
References

1. NVMe/NVMe-MI - [http://nvmexpress.org/](http://nvmexpress.org/)

Don’t Miss the Next Webcast!

Join us to learn about the evolution of the NVMe storage protocol and what’s in store for its future, in 2018 and beyond in our next webcast titled:

The Evolution and Future of NVMe

Tuesday, December 19th at 9:00am PT / 12:00pm ET.

https://www.brighttalk.com/webcast/12367/290529

David Allen, NVMe Board Member and Seagate’s Senior Director of Marketing

Dr. J Metz, Board Member, and R&D Engineer for the Office of the CTO for Cisco
Questions?

Visit [www.nvmexpress.org](http://www.nvmexpress.org) for more information on NVM Express technology

Follow us:

Twitter: [https://twitter.com/NVMexpress](https://twitter.com/NVMexpress)
LinkedIn: [https://www.linkedin.com/company/11106843/](https://www.linkedin.com/company/11106843/)
Facebook: [https://www.facebook.com/NVM-Express-137395516813022/](https://www.facebook.com/NVM-Express-137395516813022/)