UNH–IOL
NVMe Testing Service

Test Plan for NVMe-MI Conformance
Version 25.0
Target Specification: NVMe-MI 2.1
Technical Document

University of
New Hampshire
Interoperability Labs

Last Updated: February 05, 2026

UNH–IOL NVMe Testing Service
21 Madbury Rd Suite 100
Durham, NH 03824

Tel: +1 603–862–0090
Fax: +1 603–862–4181
Email: nvmelab@iol.unh.edu

TABLE OF CONTENTS

MODIFICATION RECORD

2017 February 7 (Version 7.0 r01) Initial Release
David Woolf:

2017 February 14 (Version 7.0 r02) Initial Release
David Woolf: Adjusted Group 1, 2, 3 to be FYI.

2017 February 21 (Version 7.0 r03) Initial Release
David Woolf: Adjusted Group 9, 10, 11, 12 to be In Progress.

March 22, 2017 (Version 7.0)
David Woolf: Final version published to UNH-IOL site ahead of May 2017 NVMe Plugfest #7.

April 4, 2017 (Version 7.0a)
David Woolf: Updated tests 4.7, 4.10, 5.3, 7.3 to be FYI for May 2017 NVMe Plugfest #7 per direction of NVMe-MI WG.

August 29, 2017 (Version 8.0)
David Woolf: Merged original Test Specification v0.9 from NVMe-MI Working Group into this document. Made Test 8.5 FYI.

September 12, 2017 (Version 8.0a)
David Woolf: Added Appendix D on Test Tool versions supporting the tests described in this test document.

September 19, 2017 (Version 9.0 draft)
David Woolf:

- Edited procedure and observable results for Test 8.5 to accommodate new requirements published in NVMe-MI 1.0 ECN 003.

February 1, 2018 (Version 9.0 draft)
David Woolf:

- Edited observable results for Test 4.8 to accommodate for varied error responses allowed by chapter 4.2 of the NVMe-MI 1.0 specification.
- Updated tests 4.10 and 5.3 from FYI to Mandatory.
- Modified Test 8.5 to include separate cases for devices which have or have not implemented NVMe-MI 1.0 ECN 003.

August 30, 2018 (Version 10.0 release)
David Woolf:

- Release for 10.0.

November 26, 2018 (Version 11.0 release)
David Woolf:

1. Updated Test 8.5 Case 1 and Test 8.5 Case 2 to accommodate NVMe-MI 1.0 ECN 003 changing the RENT field from being a zeroes based value to not being a zeroes based value. Products implementing NVMe-MI 1.0 may or may not implement NVMe-MI 1.0 ECN 003. Products implementing NVMe-MI 1.0a or higher must implement NVME-MI 1.0 ECN 003.
2. Fixed formatting for sub test cases in Test 8.7.

April 29, 2019 (Version 12.0 release)
David Woolf:

- Updated Test 7.3 to reflect proper use of the CESF bit, per TP2008.

- Updated section on NVMe Integrators List requirements to match the NVMe Integrators List Policy Document.

2020 March 9 (Version 13.0 release)
David Woolf:

- Fixed typo in Test 8.5 test title, which mistakenly showed the test was FYI.
- Updated Test 4.6 and 4.9 to use a better reserved value.
- Modification to tests 1.4.2 and 1.4.4. to allow for the DUT to send a NACK response.

2020 July 21 (Version 14.0 release)
David Woolf:

1. Test Cases 6.3 added to address TP 6010, Command Initiated Auto Pause requirements.
2. Test Case 7.4 updated to address NVME-MI v1.0 ECN 002 requirements around Replay and the SOM field.
3. Test Case 7.5 updated to address NVME-MI v1.0 ECN 002 requirements around Replay and the SOM field.
4. Test Case 8.5.3 added to address NVME-MI v1.0 ECN 003 requirements around filtering Controller Health Status Poll responses according to Controller Selection types.
5. Test Case 8.5.4 added to address NVME-MI v1.0 ECN 003 requirements around filtering Controller Health Status Poll responses according to error selection types.
6. Test Case 8.6 updated address TP 6007 updates.
7. Test Cases 8.8.1, 2, 3, 4, 5 added to address TP 6001 Management Endpoint Buffer Read Command requirements.
8. Test Cases 8.9.1, 2, 3, 4 added to address TP 6001 Management Endpoint Buffer Write Command requirements.
9. Test Cases 8.10.1, 2, 3, 4 added to address TP 6001 SES Receive Command requirements.
10. Test Cases 8.11.1, 2 added to address TP 6001 SES Send Command requirements.
11. Test Cases 8.12.1, 2, 3 added to address VPD Read requirements.
12. Test Cases 8.13.1, 2, 3 added to address VPD Write requirements and TP 6004.
13. Test Case 9.1 added to address Identify Command over NVMe-MI for NVMe Storage Device
14. Test Case 9.2 added to address Get Features Command over NVMe-MI for NVMe Storage Device
15. Test Case 9.3 added to address Get Log Page Command over NVMe-MI for NVMe Storage Device
16. Test Cases 9.4.1, 2 added to address handling of NVMe Admin commands prohibited for NVMe Enclosures.
17. Test Cases 9.5.1 added to address handling of NVMe-MI commands during a Sanitize operation.
18. Test Case 10.3 updated to address NVME-MI v1.0 ECN 003 requirements around proper to incorrect Set Features requests for the Namespace Metadata feature.
19. Test Case 10.4 added to address NVME-MI v1.0 ECN 003 requirements around proper to incorrect Set Features requests for the Controller Metadata feature.
20. Test Case 11.1 updated to address NVME-MI v1.0a ECN 001 requirements around Product Info Area field and the CPIA field.
21. Test Case 11.2 added to address TP 6003a and Topology Multirecord requirements.
22. Test Cases 11.3 and 11.4 added to address TP 6006 PCIe Port Numbering
23. Fixed typo where Observables for 8.4 and 8.6 were swapped.

2021 May 3 (Version 15.0 release)
David Woolf:

- Program Revision Update.

2021 September 23 (Version 16.0 release)
David Woolf:

1. Updated Test 9.1.1 per requirements in NVMe-MI TP 6016b that the NVMe Storage Device (NVMESD) bit in the NVM Subsystem Report (NVMSR) field of the Identify Controller data structure shall be set to '1' for an NVMe Storage Device and the NVMe Enclosure (NVMEE) bit shall be set to '1' for an NVMe Enclosure. This test also checks that the NVMESD bit, the NVMEE bit, or both the NVMESD bit and the NVMEE bit shall be set to '1'. This test also checks if the VPD Write Cycle Remain Valid bit is cleared to 0, then the VPD Write Cycles Remains field shall be cleared to 0h.
2. Updated Test 9.3.1 per requirements in NVMe-MI TP 6016b that for the Set and Get Features commands, all Feature Identifiers supported shall be supported if received in-band on an NVMe controller or received out-of-band on a Management Endpoint.
3. Added Test 9.3.2 per requirements in NVMe-MI TP 6016b that Get Features commands for the Host Metadata FIDs (7Dh, 7Eh, 7Fh), has the SEL field set to 011b (i.e., Supported Capabilities), then the Saveable bit in Dword 0 of the corresponding completion queue entry shall be cleared to '0', and the Changeable bit in Dword 0 of the corresponding completion entry shall be set to 1.
4. Added Test 9.3.3 per requirements in NVMe-MI TP 6016b that Get Features commands for the Host Metadata FIDs (7Dh, 7Eh, 7Fh), if Generate Default Host Metadata (GDHM) field is cleared to '0' then the device shall not generate any vendor specific strings for the Element Types of the specified Host Metadata feature.
5. Added Test 9.3.4 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata FIDs (7Eh, 7Fh), with the Element Action field is cleared to 00b (Add/Replace Entry) and if the Metadata Element Descriptor with the specified Element Type does not exist in the specified Host Metadata Feature value, then the Controller shall create the descriptor in the specified Host Metadata Feature value with the value in the Host Metadata data structure.
6. Added Test 9.3.5 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata FIDs (7Eh, 7Fh), with the Element Action field is cleared to 00b (Add/Replace Entry) and one Metadata Element Descriptor with the specified Element Type exists in the specified Host Metadata Feature value, then the Controller shall replace with the value in the specified Host Metadata data structure.
7. Added Test 9.3.6 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata FID 7Dh, if the Element Action field is cleared to 00b (Add/Replace Entry) and the Feature Identifier field is set to Enhanced Controller Metadata, then the controller shall abort the Set Features command with Invalid Field in Command status and shall not change any Host Metadata Feature value.
8. Added Test 9.3.7 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata features, that if the Element Action field is set to 01b (Delete Entry Multiple), then the Controller shall delete all the specified Metadata Element Descriptors from the specified Host Metadata Feature value, if any. If none of the specified Metadata Element Descriptors are present in the specified Host Metadata Feature value, then the controller shall complete the Set Features command with a status of Successful Completion and shall not change any Host Metadata Feature value.
9. Added Test 9.3.8 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata features, that if the Element Action field is set to 10b (Add Entry Multiple), the Feature Identifier field is set to Enhanced Controller Metadata, and no Metadata Element Descriptor with the specified Element Type exists in the Enhanced Controller Metadata Feature value, then the controller shall create new Metadata Element Descriptors in the Enhanced Controller Metadata Feature value with the Element Type and the value specified in the Host Metadata data structure.
10. Added Test 9.3.9 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata features, that if the Element Action field in a Set Features command, if the Element Action field is set to 10b (Add Entry Multiple), the Feature Identifier field is set to Enhanced Controller Metadata, and one or more Metadata Element Descriptors with the specified Element Type exists in the Enhanced Controller Metadata Feature value, then the controller shall add the specified Metadata Element to the Enhanced Controller Metadata Feature value and shall not modify any existing Metadata Element Descriptors.

11. Added Test 9.3.10 per requirements in NVMe-MI TP 6016b that for Set Features commands for the Host Metadata features, that if the Element Action field is set to 10b (Add Entry Multiple) and the Feature Identifier field is not set to Enhanced Controller Metadata, then the controller shall abort the Set Features command with status Invalid Field in Command and shall not change the Host Metadata Feature value.
12. Added Test 9.3.11 per requirements in NVMe-MI TP 6016b that for a Set Features command that is submitted for a Host Metadata Feature, and host software attempts to add or replace a Metadata Element that causes the Host Metadata Feature value of the specified feature to grow larger than 4 KiB, then the controller shall abort the command with an Invalid Field In Command.
13. Added Test 9.3.12 per requirements in NVMe-MI TP 6016b that the default value for the Number of Metadata Element Descriptors of the Enhanced Controller Metadata Feature shall be 0h on a Controller Level Reset.
14. Added Test 9.3.13 per requirements in NVMe-MI TP 6016b that if a Get Features command with the SEL field set to 011b (i.e., Supported Capabilities) with the Enhanced Controller Metadata Feature value is submitted, then the NS Specific bit in Dword 0 of the corresponding completion queue entry shall be cleared to '0'.
15. Added Test 9.3.14 per requirements in NVMe-MI TP 6016b that if a Get Features command with the SEL field set to 011b (i.e., Supported Capabilities) with the Namespace Metadata Feature value is submitted, then the NS Specific bit in Dword 0 of the corresponding completion queue entry shall be set to '1'.
16. Test 11.5 added per NVMe v1.1 TP 6013, to check requirements that each NVMe Storage Device FRU shall have a FRU Information Device with a size of 256 to 4096 bytes which contains the VPD. This test also check that if the NVMe Storage Device contains an SMBus/I2C port, then the SMBus/I2C Address Info field in the FRU Information Device Element Descriptor is correct, if no SMBus/I2C port is present, then this field shall be cleared to 0h.
17. Test 11.6 added per NVMe v1.1 TP 6013, to check that FRU Information Device can be accessed by I2C/SMBus Read. This test also checks that the SMBus/I2C Address Info field in the FRU Information Device Element Descriptor is correct.
18. Test 11.7 added per NVMe v1.1 TP 6013, to check requirements that updating the VPD by writing to the FRU Information Device using I2C Writes shall not be supported if the VPD Write command is supported.
19. Test 11.8 added per NVMe v1.1 TP 6013, to check requirements that if an I2C Read is issued, then data is returned from the internal offset within the FRU Information Device and then the internal offset is incremented by 1h.
20. Test 12.1 updated per NVMe-MI V1.1 ECN 002, to check requirements that a reset of a Management Endpoint in an NVM Subsystem shall not affect any other Management Endpoint in the NVM Subsystem or any other NVM Subsystem entity.
21. Test 12.2 added per NVMe-MI v1.1 TP 6015 regarding SMBus Reset Enhancements to check that an SMBus Reset shall be treated by each Command Slot in the SMBus/I2C Management Endpoint as if an implicit Abort Control Primitive was received with the exception that the Management Endpoint does not transmit the Abort Control Primitive Response Messages.

2022 January 21 (Version 17.0 release)
Tim Sheehan:

- Program Revision Update.

2022 July 14 (Version 18.0 release)
Tim Sheehan:

1. Test 7.7 added per NVMe-MI v1.1 TP6020a.
2. Test 9.2.2 added per NVMe-MI v1.1 TP6024
3. Test 7.6 added per NVMe-MI V1.2 TP6029
4. Tests 8.8.3&4, 8.9.3&4, 8.10.2&3, 4.7, 8.12.3, and 4.8 modified per NVMe-MI V1.2 TP6029
5. Tests 11.1-8 modified version check

2023 January 04 (Version 19.0 release)
Tim Sheehan:

1. Test 9.2.3 added per TP6026
2. Tests 12.3 & 12.4 added per TP6027
3. Test 9.3.15 added per TP6025a
4. Test 9.6 added per TP6030

2023 July 19 (Version 20.0 release)
Tim Sheehan:

1. Tests 10.3 case 1 & 2 updated from (FYI) to (M)
2. Tests 10.4 case 1 & 2 updated from (FYI) to (M)
3. Modified Test 7.6 to verify Msg tag in Relayed Response Message
4. Modified Test 7.4 & 7.5 to include version check for 1.4 or less support
5. New Tests 7.8 & 7.9 For Replay response message
6. Modified Tests 11.4-8 and 12.1&2 to FYI, we incorrectly labeled M
7. Modified Test 11.3&4 per requirements in TP6017 VPD Data structure enhancements and TP6019a Device Form Factor Table
8. New Test 7.10 per requirements in TP4136 Should to Shall
9. New Tests 7.8 & 9, 9.3.16 per requirements in TP6022 MI Maintenance
10. Modified Tests 4.8, 8.13.1-3, 11.7, 9.5.1, & 9.4.1&2 per requirements in TP6028 MI Command Optional Support
11. Modified Test 9.6.1 per requirements in TP6031 Progress Detect During Paused Process State
12. New Test 9.2.4 Get Log Page DOFST & DLEN

2024 January 18 (Version 21.0)
Carter Snay:
Tests with Status Changes:

1. No tests moved from FYI to Mandatory in this test plan revision.

New and Modified test cases

1. Added new test group 13 “Shutdown Processing” cases 1-7 based on requirements of TP6032.
2. Modified test case 9.6.1 by adding a third observable result to specify that only 2 “More Processing Required Response” for a given Command Message are sent as per requirements in TP6033.
3. Added tests 7.11-7.17 to test requirements in TP6033, primarily testing the pause flag enhancements.
4. Added new tests 12.5-12.16 to address new Reset clarification requirements set in TP6027b
5. Updated References to NVMe Management Interface Specification 1.2d

July 29, 2024 (Version 22.0)
Carter Snay:
Tests with Status Changes:

1. Test 8.5.3 updated from (FYI) to (M)
2. Test 8.5.4 updated from (FYI) to (M)
3. Test 8.12 all cases updated from (FYI) to (M)
4. Test 8.13 all cases updated from (FYI) to (M)
5. Test 9.1 updated from (FYI) to (M)
6. Test 9.5 updated from (FYI) to (M)

New and Modified test cases

1. Modified test case 9.2.2 to verify that the RAE bit is ignored to address requirements in ECN119
2. Added test cases 3 and 4 to test 8.6 to test requirements from ECN119
3. Modified test cases 12.3-6, 12.8, 12.9, 12.14, 12.15, 4.3, and 8.5.1-4 to account for multiple Management Endpoints as required by TP6034. Also 12.8 had modifications to address requirements from TP6021.
4. Added new test cases 4.7.2, 4.8.2 and 4.9.2, which expects a higher VPD size limit as dictated by NVMe-MI 1.2 and added a version check to 4.7.1, 4.8.1 and 4.9.1.

5. Added new test cases to test 7.3 NVMe-MI Get State Primitive Response, cases 4, and 5, and new tests cases to test 8.5 NVMe-MI Controller Health Status Poll cases 5 and 6 to test requirements from TP6021 Status Reporting Enhancements.
6. Modified test 9.4 to allow for devices supporting NVMe-MI version 1.2 to return Invalid Command Opcode as a response.
7. Added new group and 54 test cases for Asynchronous Event requirements as defined in TP6035 and TP6035a

2025 January 23 (Version 23.0)

Carter Snay :

New and Modified test cases

1. Updated references to SMBus/I2C to 2 Wire Port in tests 4.4, 12.1, 12.2, 12.3, 12.14, 12.15 and updated test case name accordingly. Semantic change, non-functional update for TP6037.
2. New tests 15.1 and 15.2 to test for I3C modifications made in TP6037.
3. Moved test 12.10 to case 12.10.1, and added 12.10.2 to test for NVMe-MI command servicing while a CLR is being performed on another controller in the NVM Subsystem per TP6038
4. New test 11.9.1 and 11.9.2 to verify Wire Port VPD/Mux Access is NACKed if not available to the host, per TP6038.
5. New test 8.12.3 to verify timeout values returned from a VPD are followed per TP6038
6. Updated test cases 8.8.3 and 8.9.3 by modifying the check of DOFST > than the MEB size to be ‘greater than or equal to’ due to a change in NVMe MI 1.2 and 2.0 specification change, and changed the observable results to ensure the PEL indicates the DOFST field and not the DLEN field.
7. Updated possible problems for test case 4.7, 4.8, and 4.9 to add additional information about VPD size changes in NVMe-MI v1.2.

2025 July 31 (Version 24.0)

Carter Snay :

Tests with Status Changes:

1. Test 7.6 updated from (FYI) to (M)
2. Test 9.1.1 updated from (FYI) to (M)
3. Test 9.5 updated from (FYI) to (M)
4. Test 11.2 updated from (FYI) to (M)
5. Test 11.4 updated from (FYI) to (M)
6. Test 11.5 updated from (FYI) to (M)

New and Modified test cases

1. New test cases 9.2.5, 9.3.17, and 9.3.18 added per requirements in TP4094a Management Network Addresses
2. Modified test case 4.7.2 step 9 to change the command from a VPD Write to a VPD Read
3. New test 9.7 to test DOFST and the Admin Command expected return status based on clarifications in ECN122
4. Modified test 9.3.5 to test new clarifications on the DLEN field limits from ECN122
5. New test 9.3.18 to verify that when the host issues an NVMe Admin Command Request with bits 1:0 of the DLEN field set to a non-zero value, the device returns an Invalid Parameter Error Response with the PEL field indicating the DLEN field as per ECN122.
6. Modified test 8.9.3 to change the procedure from sending a command with DOFST => Management Endpoint Buffer Size to DOFST = Management Endpoint Buffer Size (removed the greater)

2026 February 5 (Version 25.0)

Carter Snay :

New and Modified test cases

1. Modified test 4.7.2 to account for increased FRU size in NVMe-MI 2.0.
2. Removed CID 04 from the test procedures of 4.2, 4.4, and 8.2, as this is a reserved CID and was causing false failures that required passing with comments.

3. Updated tests that reference MJR and MNR versions 1.2 to “NVMe-MI version 1.2 or later...” to allow for NVMe-MI 2.0 specification versions. Previously, it was only limited to MJR version 1.

ACKNOWLEDGMENTS

The UNH–IOL would like to acknowledge the efforts of the following individuals in the development of this test plan:

Gordon Getty	Teledyne-LeCroy
David Woolf	UNH-IOL
Tim Sheehan	UNH-IOL
Carter Snay	UNH-IOL
Timofei Nikshych	UNH-IOL
William Lee	UNH-IOL
Mitchell Tilton	UNH-IOL
Austin Snow	UNH-IOL

INTRODUCTION

The University of New Hampshire's InterOperability Laboratory (IOL) is an institution designed to improve the interoperability of standards–based products by providing a neutral environment where a product can be tested against other implementations of a common standard, both in terms of interoperability and conformance. This particular suite of tests has been developed to help implementers evaluate the NVMe-MI functionality of their products. This test suite is aimed at validating products in support of the work being directed by the NVMe Organization.

These tests are designed to determine if a product conforms to specifications defined in the NVMe Management Interface specification, hereafter referred to as the "NVMe-MI Specification"). Successful completion of these tests provide a reasonable level of confidence that the Device Under Test (DUT) will function properly in many NVMe-MI environments.

REFERENCES

The following documents are referenced in this text:

1. NVM Express Management Interface 1.1d, March 11 2021
2. NVM Express Management Interface 1.2d January 10 2024
3. NVM Express Management Interface 2.0 August 5 2024

Group 1: MCTP Base Tests

Test 1.1 – MCTP Endpoint ID – (FYI)

Purpose: Source Endpoint ID should be set to the assigned value

References:
MCTP Base Specification Section 8.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Endpoint ID MCTP command
3. Wait for response.

Observable Results:
1. Ensure that the Endpoint ID returned by Get Endpoint ID command is the one assigned during MCTP initialization. f returned correctly, then result is PASS, otherwise, result is FAIL

Possible Problems: None

Test 1.1.1 – MCTP Endpoint ID Response – Reserved Bits – (FYI)

Purpose: The intent of this test is to ensure that all the reserved bits of the response are set to zero in the Get Endpoint ID Response

References:
MCTP Base Specification Section 8.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get EndPointID command
3. Wait for the Response Message

Observable Results:
1. Ensure that the Get Endpoint ID command returns with successful completion.
2. Ensure that the following bits are reserved:
    a. Bits 7:6 of byte 3 are reserved.
    b. Bits 2:0 of byte 3 are reserved.
    c. Endpoint Type (Bits 5:4 of byte 3) have values 10b and 11b reserved.

Possible Problems: None

Test 1.2 – MCTP Packet Sequence Number – (FYI)

Purpose: After the SOM packet, the packet sequence number must increment modulo 4 for each subsequent packet belonging to a given message up through the packet containing the EOM flag.

References:
MCTP Base Specification Section 8.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Transmission Unit Size
3. Wait for Response Message
4. Execute NVMe-MI VPD Read with Data Offset = 0 and Data Length = 256.
5. Wait for Response

Observable Results:
1. Ensure that the Get Endpoint ID command returns with successful completion.
2. Ensure that the Get Transmission Unit Size returns the default value of 64
3. Ensure that the response of VPD Read is split into 5 MCTP packets.
4. Ensure that the sequence number of the packets are a modulo of 4 and are in order.

Possible Problems: None

Test 1.3 – MCTP Tag Owner and Message Tag Bits – (FYI)

Purpose: For messages that are split up into multiple packets, the Tag Owner (TO) and Message Tag bits remain the same for all packets from the SOM through the EOM.

References:
MCTP Base Specification Section 8.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Transmission Unit Size
3. Wait for Response Message
4. Execute NVMe-MI VPD Read with Data Offset = 0 and Data Length = 256.
5. Wait for Response;

Observable Results:
1. Ensure that the Get Endpoint ID command returns with successful completion.
2. Ensure that the Get Transmission Unit Size returns the default value of 64
3. Ensure that the Tag Owner and Message Bits of the VPD Read response MCTP packets are the same,
1. Starting from the First message (SOM = 1, EOM = 0) to the last packet (SOM = 0, EOM = 1).

Possible Problems: None

Test 1.4.1 – MCTP Bad Packet 1 – (FYI)

Purpose: Individual packets are dropped (silently discarded) by an endpoint under the following conditions. These packets are discarded before being checked for acceptance or rejection for message assembly. Therefore, these packets will not cause a message assembly to be started or terminated.
Test case 1.4.1: Unexpected “middle” packet or “end” packet will be properly handled

References:
MCTP Base Specification Section 8.6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Endpoint ID command, wait for response
3. Send MCTP packet (Get Endpoint ID command) with SOM and EOM not set ("middle" packet)
4. Wait for 126 milliseconds (MT2, response timeout)
5. Execute Get Endpoint ID command, wait for response
6. Send MCTP packet (Get Endpoint ID command) with SOM not set and EOM set ("end" packet)
7. Wait for 126 milliseconds (MT2, response timeout)
8. Execute Get Endpoint ID command, wait for response

Observable Results:
1. Ensure that the Get Endpoint ID commands with "middle" packet designation is silently dropped by the DUT.
2. Ensure that the Get Endpoint ID commands with "end" packet designation is silently dropped by the DUT.
3. Ensure that the proper responses for correct Get Endpoint ID commands are returned by the DUT.
4. If all of the above true, then result is PASS, otherwise result is FAIL

Possible Problems: None

Test 1.4.2 – MCTP Bad Packet 2 – (FYI)

Purpose:
Bad Message Integrity Check – Ensure that all Get EndPointID requests with a bad Message Integrity Check are dropped silently by the DUT
These packets are discarded before being checked for acceptance or rejection for message assembly.
Therefore, these packets will not cause a message assembly to be started or terminated.

References:
MCTP Base Specification Section 8.8

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Configuration Get command
3. Wait for the Response Message
4. Execute Configuration Get with a wrong Message Integrity Check
5. Allow for a NACK response message from the DUT.
6. Execute Configuration Get command
7. Wait for the Response Message

Observable Results:
1. Ensure that the Configuration Get returns a successful completion.
2. Ensure that the Configuration Get with wrong message integrity check is dropped silently.
3. Ensure that the Last Configuration Get returns a successful completion.

Possible Problems: None

Test 1.4.3 – MCTP Bad Packet 3 – (FYI)

Purpose: Individual packets are dropped (silently discarded) by an endpoint under the following conditions. These packets are discarded before being checked for acceptance or rejection for message assembly. Therefore, these packets will not cause a message assembly to be started or terminated.
Test 1.4.3: Bad, unexpected, or expired message tag is handled correctly.

References:
MCTP Base Specification Section 8.6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Endpoint ID command, wait for response
3. Send MCTP packet (Get Endpoint ID command) with TO bit set to zero
4. Wait for 126 milliseconds (MT2, response timeout)
5. Execute Get Endpoint ID command, wait for response

Observable Results:
1. Ensure that the Get Endpoint ID command with bad tag owner is silently dropped by the DUT.
2. Ensure that the proper responses for correct Get Endpoint ID commands are returned by the DUT. If all of the above true, then result is PASS, otherwise result is FAIL

Possible Problems: None

Test 1.4.4 – MCTP Bad Packet 4 – (FYI)

Purpose: Individual packets are dropped (silently discarded) by an endpoint under the following conditions. These packets are discarded before being checked for acceptance or rejection for message assembly. Therefore, these packets will not cause a message assembly to be started or terminated.
Test 1.4.4: Unknown Destination EID.

References:
MCTP Base Specification Section 8.6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get EndPointID command
3. Wait for the Response Message
4. Repeat Get Endpoint Id command with Destination EID set to 0x11 (invalid), 0x22 (invalid), 0x33 (invalid),0xAB (valid)
5. Allow for a NACK response message from the DUT.

Observable Results:
1. Ensure that the Get Endpoint ID command returns with successful completion.
2. Ensure that the Get Endpoint IDs with a different Destination EPID is dropped silently by the DUT.
3. Ensure that the last Get Endpoint ID command returns a successful completion

Possible Problems: None

Test 1.4.5 – MCTP Bad Packet 5 – (FYI)

Purpose: Individual packets are dropped (silently discarded) by an endpoint under the following conditions. These packets are discarded before being checked for acceptance or rejection for message assembly. Therefore, these packets will not cause a message assembly to be started or terminated.
Test 1.4.5: Bad header version is handled correctly

References:
MCTP Base Specification Section 8.6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Endpoint ID command, wait for response
3. Send MCTP packet (Get Endpoint ID command) with bad header version (set to 2 instead of 1)
4. Wait for 126 milliseconds (MT2, response timeout)
5. Execute Get Endpoint ID command, wait for response

Observable Results:
1. Ensure that the Get Endpoint ID command with bad header version is silently dropped by the DUT.
2. Ensure that the proper responses for correct Get Endpoint ID commands are returned by the DUT.
3. If all of the above true, then result is PASS, otherwise result is FAIL

Possible Problems: None

Group 2: MCTP Control Message Tests

Test 2.1 – MCTP Control Instance ID – (FYI)

Purpose: Instance ID in the MCTP Response message should be set to the same value as in the MCTP Request message

References:
MCTP Base Specification Section 10.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get EndPointID command
3. Wait for the Response Message
4. Execute Get EndPointID command with instance ID set to 0x6, 0xC, 0x11, 0x1F

Observable Results:
1. Ensure that the Get Endpoint ID command returns with successful completion.
2. Ensure that the Datagram Bit is set to 0 and that the instance ID of the Get EndPointID Request and the Instance id of the Get EndPointID of the Response are the same.
3. Repeat the same for all instance IDs.

Possible Problems: None

Group 3: MCTP Commands

Test 3.1 – MCTP Command Set Endpoint ID – (FYI)

Purpose: The intention of this test is to ensure that proper Completion Messages are returned for mandatory MCTP Command Control Messages

References:
MCTP Base Specification Section 11.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization

Observable Results:
1. Ensure that the response to MCTP Command Set Endpoint ID returns with successful completion.
2. Ensure that Bits 7:6 of byte 2 are reserved
3. Ensure that Bits 5:4 of byte 2 of response can only have 00b and 01b as values, 10b and 11b are reserved.
4. Ensure that Bits 3:2 of byte 2 are reserved.
5. Ensure that Bits 1:0 of Byte 2 can only have 00b, 10b, and 01b as values, 11b is reserved.
6. Ensure that if Endpoint ID Assignment Status == 00b, EID Setting (byte 3)= = 0xAB

Possible Problems: None

Test 3.2 – MCTP Command Get MCTP Version – (FYI)

Purpose: The intention of this test is to ensure that proper Completion Messages are returned for mandatory MCTP Command Control Messages

References:
MCTP Base Specification Section 11.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Perform MCTP Get Version Support with Message Type Number set to 0xFF (MCTP base specification Version information).
3. Wait for Response.
4. Perform MCTP Get Version Support with Message Type Number set to 0x00(MCTP Control Protocol Message Version Information).
5. Wait for Response.
6. Perform MCTP Get Version Support with Message Type Number set to 0x10(MCTP Control Protocol Message Version Information).
7. Wait for Response.
8. Perform MCTP Get Version Support with Message Type Number set to 0x70(MCTP Control protocol message version information)
9. Wait for Response.

Observable Results:
1. Ensure that the response to MCTP Command Get MCTP version support returns with successful completion and return Message Type not supported if Completion code is 0x80.
2. Ensure that the number of Version number entries (32 bit version numbers) is equal to the Version Number entry Count (one based).
3. Repeat the above for Message Type Number = 0x00.
4. Ensure that for Message Type Number = 0x10 and 0x70, the Completion Code in Response Data is set to 0x80.

Possible Problems: None

Test 3.3 – MCTP Command Get Message Type – (FYI)

Purpose: The intention of this test is to ensure that proper Completion Messages are returned for mandatory MCTP Command Control Messages

References:
MCTP Base Specification Section 11.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Perform MCTP Get Message Type Support.
3. Wait for Response.

Observable Results:
1. Ensure that the response to MCTP Command Get Message Type returns with successful completion.
2. Ensure that the Number of bytes in the Message Data corresponds to the MCTP Message Type Count in the Response Data.ie.: MCTP Message Type Count = (Length of Response Data) - 1 .

Possible Problems: None

Test 3.4 – MCTP Command Prepare for Endpoint Discovery – (FYI)

Purpose: The intention of this test is to ensure that proper Completion Messages are returned for mandatory MCTP Command Control Messages

References:
MCTP Base Specification Section 11.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.

Observable Results:
1. Ensure that the response to MCTP Command Prepare for EndPoint Discovery returns with successful completion.

Note: SMBus should return ERROR_UNSUPPORTED_CMD as Completion Code.

Possible Problems: None

Test 3.5 – MCTP Command Endpoint Discovery – (FYI)

Purpose: The intention of this test is to ensure that proper Completion Messages are returned for mandatory MCTP Command Control Messages

References:
MCTP Base Specification Section 11.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.

Observable Results:
1. Ensure that the response to MCTP Command EndPoint Discovery returns with successful completion.

Possible Problems: None

Test 3.6 – MCTP Command Get Endpoint ID – (FYI)

Purpose: The intention of this test is to ensure that proper Completion Messages are returned for mandatory MCTP Command Control Messages

References:
MCTP Base Specification Section 11.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization

Observable Results:
1. Ensure the response returns with Successful Completion and that all the reserved bits of the response are set to zero in the Get EndPointID Response.

Possible Problems: None

Group 4: NVMe Error Handling

Test 4.1 – NVMe-MI Invalid Opcode – (Mandatory)

Purpose: Invalid Command Opcode Status response should be sent when the request Opcode is set to 08h – BFh

References:
NVMe-MI Base Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 0, wait for response
3. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 1, wait for response
4. Perform Management Interface Command with opcode 0x08 to the Management Endpoint Command Slot 0, wait for response
5. Perform Management Interface Command with opcode 0x29 to the Management Endpoint Command Slot 1, wait for response
6. Perform Management Interface Command with opcode 0x5A to the Management Endpoint Command Slot 0, wait for response
7. Perform Management Interface Command with opcode 0xBE to the Management Endpoint Command Slot 1, wait for response

Observable Results:
1. Ensure that for the Configuration Get commands the Response Messages are returned properly.
2. Ensure that for the commands with reserved opcodes the Response Messages are returned with Invalid Command Opcode Status.
3. If returned correctly, then result is PASS, otherwise, result is FAIL

Possible Problems: None

Test 4.2 – NVMe-MI Reserved Identifier – (Mandatory)

Purpose: Configuration Get: Specifying a reserved identifier in the Configuration Identifier field causes the command to complete with an Invalid Parameter error status.

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Configuration Get Command requesting MCTP Transmission Unit Size
3. Wait for the Response message
4. Perform Configuration Get Command with Configuration Identifier set to 0
5. Wait for the Response message
6. Repeat the above for Configuration Identifier values of 0x56, 0xBF

Observable Results:
1. Ensure that for the first command proper response is returned, and status is set to Success;
2. Ensure that for the rest of the commands:
    a. Invalid Parameter Error Response is sent
    b. In the Parameter Error Location field of the response, the byte position is set to 8 and bit position - to 0.

Possible Problems: None

Test 4.3 – NVMe-MI Health Status Change – (Mandatory)

Purpose: For Health Status Change (Configuration Identifier 02h) - A Management Endpoint shall complete a Configuration Get command on this Configuration Identifier with a Success Response Message. The NVMe Management Response field is reserved and there is no Response Data.

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: August 1st, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Perform Endpoint Discovery.
3. For each Management Endpoint:
4. Execute Configuration Get with Configuration Identifier set to 02h (Health Status Change)
5. Wait for Response message
    a. Execute Configuration Get with Configuration Identifier set to 02h (Health Status Change)
    b. Wait for Response message

Observable Results:
1. For each Management Endpoint, ensure that Configuration Get command returns with successful completion.
2. For each Management Endpoint, ensure that the NVMe Management Response field is reserved and that no Response data was returned (NVMe Management Response Field - bits 31:8 of dw1 of the response).

Possible Problems: None

Test 4.4 – NVMe-MI Reserved Configuration Identifier – (Mandatory)

Purpose: Configuration Set:

- Specifying a reserved identifier in the Configuration Identifier field causes the command to complete with an Invalid Parameter error status.- SMBus/I2C Frequency (Configuration Identifier 01h). If the specified frequency is not supported or the Port Identifier specified is not an SMBus/I2C port, the Management Endpoint shall respond with an Invalid Parameter error status.
  (Can use Reserved value of 0 or 4-F)
- MCTP Transmission Unit Size (Configuration Identifier 03h). If the specified MCTP Transmission Unit Size is not supported or the Port Identifier specified is not valid, the Management Endpoint shall abort the command and send a Response Message with an Invalid Parameter error status.

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 23 2025

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization
2. (SMBus only) Perform Configuration Get to ensure SMBus is supported
3. Perform Configuration Set Command with Configuration Identifier 0x03 (MCTP Transmission Unit Size: Size = 64 bytes)
4. Wait for the Response message
5. Perform Configuration Set Command with Configuration Identifier set to 0
6. Wait for the Response message
7. Repeat the above for Configuration Identifier values of 0x56, 0xBF
8. Perform Configuration Set Command with Configuration Identifier set to SMBus/I2C Frequency and Frequency Set to 100 kHz
9. Wait for Response Message
10. Repeat the above for Frequency set to 0h, 4h, and Fh and non-2 wire port.

Observable Results:
1. Ensure that the Set Configuration command returns with successful completion.
2. Ensure that for the rest of the commands with invalid Configuration Identifier, Invalid Parameter Error Response is sent.
3. Ensure that Set Configuration - SMBus/I2C Frequency command returns with successful completion if Frequency is supported, otherwise should return Invalid Parameter response
4. Ensure that for the rest of the commands an Invalid Parameter error response is sent.

Possible Problems: None

Test 4.5 – NVMe-MI MAXRENT Error – (Mandatory)

Purpose: Controller Health Status Poll: For Maximum Response Entries (MAXRENT) in Request message. Specifying 256 entries is interpreted as an Invalid Field. Command to complete with an Invalid Parameter error status.

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: Feb 3 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Controller Health Status Poll Command with the following parameters:
    a. -Report all set:
    b. Include PCI Functions set
    c. Controller Status Changes set
    d. -MAXRENT set to 0xFF (which results in 256 entries, as this is zero-based value), all other parameters set to zero.
3. Wait for the Response message.

Observable Results:
1. Ensure that the proper response is returned, and status is set to Invalid Parameter.
2. Ensure that for the next command:
    a. Invalid Parameter Response is set
    b. In the Parameter Error Location field of the response the byte position is set to 10 and bit position - to 0.

Possible Problems: None

Test 4.6 – NVMe-MI Reserved Data Structure Type – (Mandatory)

Purpose: Read NVMe-MI Data Structure:

- If Data Structure Type (DTYP) in Request is set to reserved value, the command shall complete with an Invalid Parameter error status. (Implicit)

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: November 13, 2019

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Read NVMe-MI Data Structure Command with Data Structure Type set to a reserved value, such as 0xFF
3. Wait for the Response message
4. Repeat the above Data Structure Type values of 0x23, 0x56, 0x9A, 0xFF.

Observable Results:
1. Ensure that for each command the proper response is returned, and status is set to Invalid Parameter.
2. Ensure that for the rest of the commands:
    a. Invalid Parameter Error Response is sent
    b. In the Parameter Error Location field of the response the byte position is set to 11 and bit position - to 0.

Possible Problems: Some values that were reserved in NVMe-MI v1.0 are used in NVMe-MI 1.1, so any test implementation should pay attention to these differences.

Test 4.7 – NVMe-MI Invalid VPD Read Size -(FYI)

Purpose: VPD Read:

-If the Data Length plus Data Offset fields are greater than the size of the VPD, then the Management Endpoint does not return the VPD contents and responds with an Invalid Parameter error status response

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: NVMe-MI Invalid VPD Read Size (FYI)

Test Procedure:
1. Perform MCTP initialization
2. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
3. If the DUT is supporting NVMe-MI 1.2 or later, this test is not applicable.
4. Perform Read VPD with Data Length set to 7 bytes and Offset to 1 byte
5. Wait for the Response message
6. Perform Read VPD with Data Length set to 5 bytes and Offset to 260 bytes
7. Wait for the Response message
8. Perform Read VPD with Data Length set to 300 bytes and Offset to 5 bytes
9. Wait for the Response message

Observable Results:
1. Ensure that for First Command proper response is returned and status is set to 'success'
2. Ensure that for the rest of the commands: Invalid Parameter Error Response is returned.
3. In the Parameter Error Location field of the response the byte position is set to 8 (0x08) and bit position to 0.
    a. flagging the length field in NVMe management DWord 1 would be correct
    b. If the offset is not within range then the byte position 8 should be returned. If the offset is good, but the length causes the boundary to be exceeded then 0xc should be the byte indication

Possible Problems: Beginning with NVM Express Management Interface Specification, Revision 1.2, the FRU Information Size limit was updated to 4096 bytes, raised from 256 bytes. As a result, devices that support a VPD size > 256 bytes will fail this test case, if they are implementing specification behavior that is compliant with NVMe-MI 1.2 or greater.

Case 2: NVMe-MI Invalid VPD Read Size NVMe-MI 1.2 or Greater (FYI)

Test Procedure:
1. Perform MCTP initialization
2. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.

3. Perform Read VPD with Data Length set to 7 bytes and Offset to 1 byte
4. Wait for the Response message
5. Perform Read VPD with Data Length set to 5 bytes and offset to the length of the VPD minus 5.
6. Wait for the Response message
7. Perform Read VPD with Data Length set to 4096 bytes and Offset to 5 bytes.
8. Wait for the Response message
9. Perform Read VPD with Data Length set to 4096 bytes and Offset to 5 bytes
10. Wait for the Response message

Observable Results:

1. Ensure that for the first two commands, a proper response is returned and status is set to 'success'
2. For the next three commands, ensure that if DLEN + DOFST > FRU size, the command returns Invalid Parameter Error Response and if no error response is returned, that the status is ‘success’.
3. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates DLEN field
4. In the Parameter Error Location field of the response:
    a. If the DOFST field is greater than or equal to the FRU size then the byte position is set to 8 (0x08) and the bit position is set to 0.
    b. If the DOFST field is less than the FRU size then the byte position is set to C (0x0C) and the bit position is set to 0.

Possible Problems: None

Test 4.8 – NVMe-MI Invalid VPD Write Status – (Mandatory)

Purpose: VPD Write:
The intent of this test is to ensure that the VPD Write returns Parameter Error Response when the Data length and Data Offset Fields greater than the size of VPD.

References:
NVMe-MI Specification Chapter 4.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 17, 2023

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: NVMe-MI Invalid VPD Write Status (FYI)

Test Procedure:
1. Perform MCTP initialization
2. If the DUT is supporting NVMe-MI 1.2 or later, this test is not applicable.
3. Perform Write VPD with Data Length set to 7 bytes and Offset to 1 byte
4. Wait for the Response message
5. Perform Write VPD with Data Length set to 5 bytes and Offset to 260 bytes
6. Wait for the Response message
7. Perform Write VPD with Data Length set to 300 bytes and Offset to 5 bytes
8. Wait for the Response message

Observable Results:
1. Ensure that for First Command proper response is returned and status is set to 'success'.
2. Ensure that for the rest of the commands:
    a. Invalid Parameter Error Response is sent
    b. In the Parameter Error Location field of the response the byte position is set to 10 (0x0A) and bit position - to 0.
    c. If DOFST is out of bounds then 0x8 should be returned, If DOFST is valid and DLEN causes too large, then 0xC is returned.
3. If there are multiple error conditions detected by the DUT, then the response may be an error code other than 'Invalid Parameter Error'. This is acceptable behavior.

Possible Problems: Devices supporting NVMe-MI 1.2 or beyond, the VPD Write and Reset commands are now optional. And, beginning with NVM Express Management Interface Specification, Revision 1.2, the FRU Information Size limit was updated to 4096 bytes, raised from 256 bytes. As a result, devices that support a VPD size > 256 bytes will fail this test case, if they are implementing specification behavior that is compliant with NVMe-MI 1.2 or greater.

Case 2: NVMe-MI Invalid VPD Write Status NVMe-MI 1.2 or Greater (FYI)

Test Procedure:
Test Procedure:
1. Perform MCTP initialization

2. Perform Write VPD with Data Length set to 7 bytes and Offset to 1 byte
3. Wait for the Response message
4. Perform Write VPD with Data Length set to 5 bytes and Offset to 260 bytes
5. Wait for the Response message
6. Perform Write VPD with Data Length set to 300 bytes and Offset to 5 bytes
7. Wait for the Response message
8. Perform Write VPD with Data Length set to 4096 bytes and Offset to 5 bytes
9. Wait for the Response message

Observable Results:

1. Ensure that for Three Command proper response is returned and status is set to 'success'.
2. Ensure that for the fourth command in step 4:
    d. Invalid Parameter Error Response is sent
    e. In the Parameter Error Location field of the response the byte position is set to 10 (0x0A) and bit position - to 0.
    f. If DOFST is out of bounds then 0x8 should be returned, If DOFST is valid and DLEN causes too large, then 0xC is returned.
3. If there are multiple error conditions detected by the DUT, then the response may be an error code other than 'Invalid Parameter Error'. This is acceptable behavior.

Possible Problems: Devices supporting NVMe-MI 1.2 or beyond, the VPD Write and Reset commands are now optional. Additionally, beginning with NVM Express Management Interface Specification, Revision 1.2, the FRU Information Size limit was updated to 4096 bytes, raised from 256 bytes. As a result, devices that support a VPD size > 256 bytes will fail this test case, if they are implementing specification behavior that is compliant with NVMe-MI 1.2 or greater.

Test 4.9 – NVMe-MI Invalid Parameter Status – (Mandatory)

Purpose: Invalid Parameter Status response shall be sent when Command message contains invalid parameter. The Parameter Error Location in the Response shall be set to the proper location of the invalid parameter.

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: November 13, 2019

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: NVMe-MI Invalid Parameter Status (FYI)

Test Procedure:
1. Start with Endpoint discovery
2. If the DUT is supporting NVMe-MI 1.2 or later, this test is not applicable.
3. Send Configuration Get TU Size command to Command Slot 0 with Invalid Config Id = 0x04
4. Wait for response
5. Send NVMe-MI Read VPD with Offset set to more than 256
6. Wait for response
7. Send Read NVMe-MI Data Structure Command with invalid DTYP set to a reserved value such as 0xFF.
8. Wait for response

Observable Results:
1. Ensure that Invalid Parameter Status response shall be sent when Command message contains invalid parameter.

Possible Problems: Some values that were reserved in NVMe-MI v1.0 are used in NVMe-MI 1.1, so any test implementation should pay attention to these differences.

Case 2: NVMe-MI Invalid Parameter Status NVMe-MI 1.2 or Greater (FYI)

Test Procedure:
1. Start with Endpoint discovery
2. Send Configuration Get TU Size command to Command Slot 0 with Invalid Config Id = 0x04
3. Wait for response
4. Send NVMe-MI Read VPD with Offset set to more than 4096
5. Wait for response
6. Send Read NVMe-MI Data Structure Command with invalid DTYP set to a reserved value such as 0xFF.
7. Wait for response

Observable Results:
1. Ensure that Invalid Parameter Status response shall be sent when Command message contains invalid parameter.

Possible Problems: Some values that were reserved in NVMe-MI v1.0 are used in NVMe-MI 1.1, so any test implementation should pay attention to these differences.

Test 4.10 – NVMe-MI Invalid Command Size – (Mandatory)

Purpose: Invalid Command Size Status response should be sent when Command message contains too much/too little data.

References:
NVMe-MI Specification

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery
2. Send Configuration Get TU Size command to Command Slot 0
3. Wait for response
4. Send Configuration Get TU Size command to Command Slot 0 with Length = 12
5. Wait for response

Observable Results:
1. Ensure that Invalid Command size response shall be sent when Command message contains an Invalid Command Size.

Possible Problems: None

Group 5: NVMe Management Interface Tests

Test 5.1 – NVMe-MI Message Type – (Mandatory)

Purpose: The Message Type field specifies the type of payload contained in the message body and is required to be set to 4h in all messages associated with NVMe-MI

References:
NVMe-MI Specification Section 3.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0
3. Wait for the Response Message

Observable Results:
1. Ensure the Message type in the Response Message is set to 4.
2. If returned correctly, then result is PASS, otherwise, result is FAIL

Possible Problems: None

Test 5.2 – NVMe-MI Message IC – (Mandatory)

Purpose: All NVMe-MI messages are protected by a CRC and thus IC bit shall be set to '1' in all NVMe-MI messages.

References:
NVMe-MI Specification Section 3.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization
2. Execute NVMe-MI Configuration Get command
3. Wait for Response
4. Execute NVMe-MI Configuration Get command with IC bit set to zero
5. Wait for Response
6. Execute NVMe-MI Configuration Get command
7. Wait for Response

Observable Results:
1. Ensure that the Configuration Get command returns with successful completion.
2. Ensure that the second Get command with IC bit set to zero is dropped silently.
3. Ensure that the third Get command returns with successful completion.

Possible Problems: None

Test 5.3 – NVMe-MI CRC Check – (Mandatory)

Purpose: The Message Integrity Check field contains a 32-bit CRC computed over the contents of the NVMe-MI message. The 32-bit CRC used by NVMe-MI is CRC-32C (Castagnoli) which uses the generator polynomial 1EDC6F41h

References:
NVMe-MI Specification Section 3.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint discovery
2. Perform Configuration Get - Transmission Unit Size
3. Wait for response
4. Send Control Primitive with OpCode = Replay
5. Wait for response

Observable Results:
1. Ensure that CRC check passed

Possible Problems: None

Test 5.4 – NVMe-MI Command Slot – (Mandatory)

Purpose: For Response Messages, the CSI field indicates the Command Slot associated with the Request Message with which the Response Message is associated.
(Multi Message checking – use baseline TU Size of 64 and use VPD Read Command – up to 256 bytes)

References:
NVMe-MI Specification Section 3.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0, wait for response
3. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 1, wait for response
4. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 0, wait for response
5. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 1, wait for response.

Observable Results:
1. Ensure that for each command the CSI field in the Response Message matches the CSI field in the Request message.
2. If returned correctly, then result is PASS, otherwise, result is FAIL

Possible Problems: None

Test 5.5 – NVMe-MI MCTP packet padding – (Mandatory)

Purpose: The MCTP Transmission Unit size of the last packet in a Request Message or Response Message (i.e., the one with the EOM bit set in the MCTP header) shall be the smallest size needed to transfer the MCTP Packet Payload for that Packet with no additional padding beyond any padding required by the physical medium-specific trailer.

References:
NVMe-MI Specification Section 3.2.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Execute Get Transmission Unit Size
3. Wait for Response Message
4. Execute NVMe-MI VPD Read with Data Offset = 0 and Data Length = 256.
5. Wait for Response

Observable Results:
1. Ensure that the Get Endpoint ID command returns with successful completion during initialization.
2. Ensure that the Get Transmission Unit Size returns the default value of 64
3. Ensure that the MCTP packet with EOM = 1 in the VPD Response Message has the smallest size and that the start packet and every other middle packet is bigger than the end packet.

Possible Problems: None

Test 5.6 – NVMe-MI Message Integrity Check – (Mandatory)

Purpose: Once a complete NVMe-MI MCTP message has been assembled, the Message Integrity Check is verified. If the Message Integrity Check passes, then the message is processed. If the Message Integrity Check fails, then the message is discarded

References:
NVMe-MI Specification Section 3.2.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery
2. Send Configuration Get TUSize command to Command Slot 0
3. Wait for response
4. Send Configuration Get TUSize command to Command Slot 0
5. Wait for 150 milliseconds to exceed maximum possible latency and ensure that no response was sent
6. Send Configuration Get TUSize command to Command Slot 0
7. Wait for response

Observable Results:
1. Check that message is dropped when MIC fails

Possible Problems: None

Group 6: NVMe-MI Message Processing Tests

Test 6.1 – NVMe-MI Reserved Fields – (Mandatory)

Purpose: The purpose of this test is to check the behavior of the Response Message reserved fields

References:
NVMe-MI Specification Section 6.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0, wait for response;
3. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 0, wait for response

Observable Results:
1. Ensure that for each command the Reserved fields in the first DWord of the Response Message are set to zero.
2. If returned correctly, then result is PASS, otherwise, result is FAIL

Possible Problems: None

Test 6.2 – NVMe-MI Error Response Code – (Mandatory)

Purpose: The purpose of this test is to check that the Generic Error Response with the proper code is returned for Invalid Command Opcode (0x03), and for Invalid Command Size (0x05) and Invalid Command Input Data Size (0x06)

References:
NVMe-MI Specification Section 6.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 17, 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery
2. Send normal Configuration Get NVMe-MI command, Cmd slot 0
3. Wait for response
4. Send NVMe-MI command with undefined opcode 0x08
5. Wait for response
6. Send Configuration Get SmBusI2C Freq with Length = 12
7. Wait for response

Observable Results:
1. Check that the Generic Error Response with the proper code is returned for Invalid Command Opcode (0x03), and for Invalid Command Size (0x05) and Invalid Command Input Data Size (0x06)

Possible Problems: None

Test 6.3 – Command Initiated Auto Pause – (FYI)

Purpose: The purpose of this test is to check that the CPIA field is used properly for a NVMe-MI message.

References:
NVMe-MI Specification Section 3.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 30, 2020

Discussion: The Command Initiated Auto Pause (CIAP) bit in the Message Header of a Command Message specifies whether or not the Management Endpoint is automatically paused when a Command Message enters the Process state.

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery
2. Check Port Information Data Structure CIAPS field in the NVMe-MI Data Structure. If CIAPS is set to 0, this test is not applicable.
3. Send normal Configuration Get NVMe-MI command, with CIAP set to 1.
4. Wait for response

Observable Results:
1. Verify that the command completes successfully.

Possible Problems: None

Group 7: Control Primitives Tests

Test 7.1 – NVMe-MI Response Tag (Mandatory)

Purpose: Tag in Response matches the Tag in Request

References:
NVMe-MI Specification Section 7.2
MCTP Base Specification Section 8.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0, with Tag value 0x01, wait for response
3. Repeat the above for Endpoint Command Slot 1.

Observable Results:
1. Ensure that for each command, proper response is returned, and the Tag value in Response matches the one in Request.
2. If returned correctly, then result is PASS, otherwise, result is FAIL

Possible Problems: None

Test 7.2 – NVMe-MI Response Message (Mandatory)

Purpose: The intent of this test is to ensure that a DUT returns successful response for the Pause, Resume and Abort Control Primitives.

References:
NVMe-MI Specification Section 7.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery
2. Send Control Primitive with OpCode = Pause
3. Wait for response
4. Send Control Primitive with OpCode = Resume
5. Wait for response
6. Send Control Primitive with OpCode = Abort
7. Wait for response

Observable Results:
1. Ensure that a DUT returns successful response for the Pause, Resume and Abort Control Primitives.

Possible Problems: None

Test 7.3 – NVMe-MI Get State Primitive Response (FYI)

Purpose: Response for the GetState primitive returns the correct state based on the conditions applied. This includes the state error bits and response to Clear Error State Flags

References:
NVMe-MI Specification Section 7.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 2, 2019

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: NVMe-MI Get State Primitive Response (FYI)

Test Procedure:
1. Perform MCTP initialization
2. Execute Get EndPointID command
3. Send Get State Control Primitive Request with CESF = 1
4. Send a second Get State Control Primitive Request with CESF = 1
5. Send Get EndPoint Id with a Bad Header Version
6. Send Get State Control Primitive with CESF = 0
7. Send Get State Control Primitive with CESF = 1
8. Send Get State Control Primitive with CESF = 0
9. Repeat the above with:
10. Send Get EndPoint Id with EOM = 0 and SOM = 0 (Middle Packet)
11. Send Get EndPoint Id with a Destination Endpoint ID = 0x11
12. Send Get EndPoint Id with a Tag Owner = 01
13. Send Configuration Get - MCTP Transmission Unit Size with CRC = 0x123AB
14. Send Configuration Get - MCTP Transmission Unit Size with LCRC = 0xAB (Only for VDM)

Observable Results:
1. Perform MCTP initialization.
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform an NVM Subsystem Reset.
4. Perform a Get State Control Primitive Request to the Management Endpoint Command Slot 0, with Tag value 0x01, and Header field Byte 2, bit 0 set to 0x01 (MEB), wait for response and record all values in the returned Management Endpoint State data structure.
5. Perform a Management Endpoint Reset.
6. Perform a second Get State Control Primitive Request with the same parameters as step 3, and re-record the values in the Management Endpoint State data structure
7.)

Possible Problems: None

Case 2: Management Endpoint State Data Structure Bits Cleared on NVM Subsystem Reset (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform an NVM Subsystem Reset.
4. Perform a Get State Control Primitive Request to the Management Endpoint Command Slot 0, with Tag value 0x01, and Header field Byte 2, bit 0 set to 0x01 (MEB), wait for response and record all values in the returned Management Endpoint State data structure.
5. Perform a Management Endpoint Reset.
6. Perform a second Get State Control Primitive Request with the same parameters as step 3, and re-record the values in the Management Endpoint State data structure

Observable Results:

1. Verify that bit 14 of the Management Endpoint State data structure is set to 1.
2. Verify that bits 15 and 13:0 are cleared to 0

Case 3: Management Endpoint Data Structure Expected Error Updates (FYI)

Test Procedure:

1. Perform MCTP initialization
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform a Get State Control Primitive Request to the Management Endpoint Command Slot 0, with Tag value 0x01, and Header field Byte 2, bit 0 set to 0x01 (MEB), wait for response and record all values in the returned Management Endpoint State data structure.
4. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0 with the Clear Error State Flags bit set to 0.
5. Perform a Configuration Get Command with a Configuration Identifier of 0x1234
6. Perform a Configuration Get Command with a Configuration Identifier of 01h and set the Tag Owner field to 1b.
7. Perform 10 Configuration Get Commands with a Configuration Identifier of 01h and send the commands at random time intervals between 100 and 500ms, to simulate network latency.
8. Configure the host to send a command with SOM = 0b and EOM =0b with the MCTP packet payload size not equal to the payload size for the start packet (the packet with SOM = 1b and EOM = 0b)
9. Perform a Configuration Get Command with a Configuration Identifier of 01h and set the Destination Endpoint ID to 0xFF.
10. Perform a Configuration Get Command with a Configuration Identifier of 01h, and set the Header Version field to 0xF.
11. Perform a Configuration Get with the Configuration Identifier set to 03h to receive the MCTP Transmission Unit Size. Send a second Configuration Get with a valid Configuration Identifier but configure the MCTP Transmission Unit size to be different from what was reported.
12. Perform a Get State Control Primitive Request to the Management Endpoint Command Slot 0, with Tag value 0x01, and Header field Byte 2, bit 0 set to 0x01 (MEB), wait for response and record all values in the returned Management Endpoint State data structure.
13. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0 with the Clear Error State Flags bit set to 1. Wait for a response.

Observable Results:

1. Verify that all commands other than the Get State Control Primitive Request returned an error status
    a. Step 4 is expected to return Bad Packet or Other Physical Layer
    b. Step 5 is expected to return Unexpected or Expired Message Tag
    c. Step 6 is expected to return Out-of-Sequence Packet Sequence Number

   d. Step 7 is expected to return Incorrect Transmission Unit
   e. Step 8 is expected to return Unknown Destination Id
   f. Step 9 is expected to return Bad Header Version
   g. Step 10 is expected to return Unsupported Transmission Unit
2. Verify that bits 13:6 in the MES are set to 1
3. Verify that the values of the Management Endpoint data structure are copied to the Control Primitive Specific Response field in the Response Message in step 4
4. Verify that bits 14:03 are cleared in the Management Endpoint State data structure returned in step 13.

Case 4: NVM Subsystem Reset Clears CCSF Data Structure (FYI)

Test Procedure:
1. Perform MCTP initialization
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform a Controller Level Reset to set the Ready (RDY) bit in the CCSF data structure to 1.
4. Perform a Configuration Get on the Health Status Change (Configuration Identifier 02h). Check that at least one of the status bits in the Composite Status Flags field are set to '1'. If all bits are cleared to 0, this test is not applicable.
5. Perform a NVM Subsystem Reset.
6. Perform a second Configuration Get on Health Status Change (Configuration Identifier 02h)

Observable Results:
1. Verify that after the NVM Subsystem Reset, that all status bits except for the HwInit and the NSSRO bit in the Composite Controller Status FLAGS fields are cleared to '0'.'

Case 5: Exceed Maximum Valid Controller ID - Persistent Event Log (FYI)

Test Procedure:
1. Perform MCTP initialization
2. Perform a Controller Health Status Poll command with the Starting Controller ID (SCTLID) field set to a value larger than the maximum Controller ID in the NVM Subsystem.
3. Perform an NVMe Get Log Command with LID 0Dh to receive the Persistent Event Log.

Observable Results:
1. Verify that the Controller Health Status Poll command returned an Invalid Parameter Response with the PEL indicating SCTLID as the source of the error.

Test 7.4 – NVMe-MI Response Message Replay (Mandatory)

Purpose: Verify that DUT replays the response message according to the RRO in request, state and the rules, and that the status and RR bit in the Response control primitive is set according to the state and rules

References:
NVMe-MI Specification Section 4.4.5, 7.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 24, 2020

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:

1. This test is only applicable to a device supporting NVMe-MI 1.1 or lesser, if not skip this test
2. Perform MCTP initialization
3. Execute Get EndPointID command
4. Perform NVMe-MI Configuration Set - Transmission Unit Size
5. Wait for Response
6. Perform NVMe-MI Control Primitive Replay
7. Wait for Response

Observable Results:

1. Ensure that the Configuration Set command returns with success.
2. Ensure that the Replay Control primitive returns with success.
3. Ensure that bits 01:15 of Byte 07:08 are Reserved
4. Ensure that the Replayed Message was received successfully.
5. Verify that the first packet has SOM set and includes the Message Header of the original Response Message.

Possible Problems: None

Test 7.5 – NVMe-MI Response Replay Offset (RRO) (Mandatory)

Purpose: Verify that DUT replays the response message according to the RRO in request

References:
NVMe-MI Specification Section 4.4.5, 7.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 23, 2020

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.1 or lesser, if not skip this test
2. Perform MCTP initialization
3. Perform NVMe-MI VPD Read with Data length set to 256 Bytes.
4. Wait for Response
5. Perform NVMe-MI Control Primitive Replay with Response Replay Offset set to 2.
6. Wait for Response

Observable Results:
1. Ensure that the VPD Read command returns with success.
2. Ensure that the Replay Control primitive returns with success.
3. Ensure that the Replayed Message was received successfully with 3 packets instead of 5 because of RRO.
4. Verify that the first packet has SOM set and includes the Message Header of the original Response Message even if the Response Replay Offset is not zero.

Test 7.6 – NVMe-MI RRO > Length of Response Message (M)

Purpose: Verify that DUT attempts to replay the response message according to the RRO that is greater than the Length of the Response Message

References:
NVMe-MI Specification Section 4.2.1.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 13, 2022

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform NVMe-MI VPD Read with Data length set to 4096 or larger number of Bytes.

3. Wait for Response
4. Perform NVMe-MI Control Primitive Replay with Response Replay Offset set to a large enough to exceed the Response Message.
5. Wait for Response

Observable Results:
1. Ensure that the VPD Read command returns with success.
2. Ensure that the Replay Control primitive returns with success.
3. Verify that the Replayed Message was aborted with a status of Invalid Parameter Error Response and that the Msg tag value in the Replay Control Primitive is the same as in the replayed Response Message

Possible Problems: If this is an NVMe-MI v1.1 or lesser device it is possible to response with a different status.

Test 7.7 – NVMe-MI Get State, MEB=1 (FYI)

Purpose: Verify that DUT responds with an error when MEB =1

References:
NVMe-MI Specification Section 3.1.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 7, 2022

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Get State Control Primitive Request to the Management Endpoint Command Slot 0, with Tag value 0x01, and Header field Byte 2, bit 0 set to 0x01 (MEB), wait for response
3. Repeat the above for Endpoint Command Slot 1.

Observable Results:
1. Ensure that a DUT returns a status of Invalid Parameter Error Response

Possible Problems: None

Test 7.8 – NVMe-MI Response Message Replay, Msg Tag (FYI)

Purpose: Verify that DUT replays the response message according to the RRO in request, state and the rules, and that the status and RR bit in the Response control primitive is set according to the state and rules. The Msg tag in each packet of the replayed Response Message shall be set to the value of the Msg tag in the associated Replay Control Primitive.

References:
NVMe-MI Specification Section 4.2.1.5, 4.4.5, 7.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 07, 2023

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.2 or greater, if not skip this test
2. Perform MCTP initialization
3. Execute Get EndPointID command
4. Perform NVMe-MI Configuration Set - Transmission Unit Size
5. Wait for Response
6. Perform NVMe-MI Control Primitive Replay
7. Wait for Response

Observable Results:
1. Ensure that the Configuration Set command returns with success.
2. Ensure that the Replay Control primitive returns with success.
3. Ensure that bits 01:15 of Byte 07:08 are Reserved
4. Ensure that the Replayed Message was received successfully.
5. Verify that the first packet has SOM set and includes the Message Header of the original Response Message.
6. Verify that the Msg tag value in the Replay Control Primitive is the same as in the replayed Response Message.

Possible Problems: None

Test 7.9 – NVMe-MI Response Replay Offset (RRO), Msg Tag (FYI)

Purpose: Verify that DUT replays the response message according to the RRO in request. The Msg tag in each packet of the replayed Response Message shall be set to the value of the Msg tag in the associated Replay Control Primitive.

References:
NVMe-MI Specification Section 4.2.1.5, 4.4.5, 7.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 23, 2020

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.2 or greater, if not skip this test
2. Perform MCTP initialization
3. Perform NVMe-MI VPD Read with Data length set to 256 Bytes.
4. Wait for Response
5. Perform NVMe-MI Control Primitive Replay with Response Replay Offset set to 2.
6. Wait for Response

Observable Results:
1. Ensure that the VPD Read command returns with success.
2. Ensure that the Replay Control primitive returns with success.
3. Ensure that the Replayed Message was received successfully with 3 packets instead of 5 because of RRO.
4. Verify that the first packet has SOM set and includes the Message Header of the original Response Message even if the Response Replay Offset is not zero.
5. Verify that the Msg tag value in the Replay Control Primitive is the same as in the replayed Response Message.

Possible Problems: None

Test 7.10 – NVMe-MI Pause Primitive with CSI = 1 (FYI)

Purpose: The intent of this test is to ensure that a DUT returns the correct response for the Pause Control Primitive with the CSI bit set to 1. Verify that the Management Endpoint returns a status code of Invalid Parameter Error Response with a PEL field indicating the CSI bit.

References:
NVMe-MI 1.2 Specification Section 4.2.1.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: September 15 2023

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery
2. Send Control Primitive with OpCode = Pause, with the CSI bit set to 1.
3. Wait for response.

Observable Results:
1. Ensure that a DUT returns a status code of Invalid Parameter Error Response with the PEL field indicating the CSI bit.

Possible Problems: None

Test 7.11 – More Processing Required Response from Control Primitive (FYI)

Purpose: The intent of this test is to ensure that a DUT does not retransmit a More Processing Required Response to the Control Primitive after an operation that takes longer than the maximum Request-To-Response Time occurs

References:
NVMe-MI 1.2 Specification Section 4.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3. Otherwise, this test case is not applicable.
3. Perform a Format NVM command (DST and firmware update are other possible) on the NVMe controller. If a More Processing Required Response is received then continue to step 4, else try different commands to produce a More Processing Required Response.
4. Perform a Control Primitive, other than the Replay Control Primitive.

Observable Results:
1. Verify that the Management Endpoint does not transmit a More Processing Required Response to the Control Primitive

Test 7.12 – Pause Flag Settings (FYI)

Purpose: The intent of this test is to ensure that a DUT returns the correct response for the Pause Control Primitive with the CSI bit set to 1. Verify that the Management Endpoint returns a status code of Invalid Parameter Error Response with a PEL field indicating the CSI bit.

References:
NVMe-MI 1.2 Specification Section 4.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3. Otherwise, this test case is not applicable.
3. Perform a Format NVM command (DST and firmware update are other possible) on the NVMe controller. If a More Processing Required Response is received then continue to step 4, else try different commands to produce a More Processing Required Response.
4. Perform a Pause Control Primitive for the Command Message.
5. Perform a Resume Control Primitive for the Command Message.

Observable Results:
1. Verify that the Pause Control Primitive completes successfully.
2. Verify that the first Resume Control Primitive with CSI bit set to 1 is aborted with a status of Invalid Parameter Error Response and the PEL field set to CSI bit.
3. Verify that the second Resume Control Primitive with CSI bit set to 0, completes successfully.
4. Verify that the Command Message Completed successfully.

Test 7.13 – Format NVM, Resume Control Primitive with CSI bit set to 1, then 0 (FYI)

Purpose: The intent of this test is to ensure that a DUT returns the correct response for the Pause Control Primitive with the CSI bit set to 1. Verify that the Management Endpoint returns a status code of Invalid Parameter Error Response with a PEL field indicating the CSI bit.

References:
NVMe-MI 1.2 Specification Section 4.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery.

2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3. Otherwise, this test case is not applicable.
3. Perform a Format NVM command (DST and firmware update are other possible) on the NVMe controller. If a More Processing Required Response is received then continue to step 4, else try different commands to produce a More Processing Required Response.
4. Perform a Pause Control Primitive for the Command Message.
5. Perform a Resume Control Primitive for the Command Message with the CSI bit set to 1.
6. Perform a Resume Control Primitive for the Command Message with the CSI bit set to 0.

Observable Results:
1. Verify that the Pause Control Primitive completes successfully.
2. Verify that the first Resume Control Primitive with CSI bit set to 1 is aborted with a status of Invalid Parameter Error Response and the PEL field set to CSI bit.
3. Verify that the second Resume Control Primitive with CSI bit set to 0, completes successfully.
4. Verify that the Command Message Completed successfully.

Test 7.14 – Abort of Command Message, Pause Flag (FYI)

Purpose: The intent of this test is to ensure that when a DUT is receives an Abort Command Primitive, the pause flag is cleared to zero.

References:
NVMe-MI 1.2 Specification Section 4.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3. Otherwise, this test case is not applicable.
3. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 0, wait for response.
4. Perform a Get State Control Primitive request and wait for the response and note the Pause Flag value.
5. Perform an Abort Control Primitive for the Configuration Get MI Command Message, in Command Slot 0.
6. Perform a Get State Control Primitive request and wait for the response and note the Pause Flag value.

Observable Results:
1. Verify that all primitives and message complete successfully.
2. Verify that the Pause Flag value is cleared to zero after the Abort Control Primitive Response is successful.

Test 7.15 – Abort of Command Message, Response States (FYI)

Purpose: The intent of this test is to ensure that a DUT sets the Command Processing Abort Status correctly based on the status of the Command Slot (Idle or Receive).

References:
NVMe-MI 1.2 Specification Section 4.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:

1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3. Otherwise, this test case is not applicable.
3. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 0, wait for response.
4. Perform a Get State Control Primitive request and wait for the response.
5. Perform a Abort Control Primitive for the Configuration Get MI Command Message, in Command Slot 0.
6. Perform a Get State Control Primitive request and wait for the response.

Observable Results:

1. Verify that all primitives and messages are completed.
2. Verify that after procedure step 6, if Command Slot 0 is in the Idle state that ME responds with the CPAS field is cleared to 0.
3. Verify that after procedure step 6, if the Command Slot is in the Receive state, the ME responds with CPAS is set to 1h.
4. Verify that after procedure step 6, if the Command Slot is in the Process state, the ME is able to abort the MI Command message and CPAS field is set to 1h (before command processing) or 2h (after command processing).
5. Verify that after procedure step 6, if the Command Slot is in the Process state, but is unable to abort the Command Message, the ME will response with a response Message Status of Unable to Abort.

Possible Problem: There could be a race condition while obtaining the Get State Control Primitive values.

Test 7.16 – Replay of Command Message, Response States (FYI)

Purpose: The intent of this test is to ensure that a DUT sets the Response Replay (RR) bit correctly based on the status of the Command Slot (Transmit or Receive).

References:
NVMe-MI 1.2 Specification Section 4.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:

1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3. Otherwise, this test case is not applicable.
3. Perform Configuration Get MI Command (Cfg ID 0x1) to the Management Endpoint Command Slot 0, wait for response.
4. Perform a Get State Control Primitive request and wait for the response.
5. Perform a Replay Control Primitive for the Configuration Get MI Command Message, in Command Slot 0.
6. Perform a Get State Control Primitive request and wait for the response.

Observable Results:

1. Verify that all primitives and messages are completed.
2. Verify that after procedure step 6, if Command Slot 0 is in the Receive state that ME responds with a Response Message with success status and the RR bit is cleared to 0.
3. Verify that after procedure step 6, if the Command Slot is in the Transmit state, the ME responds with a Replay Primitive Success Response with RR is set to 1h and transmits a new Response Message

Possible Problems: There could be a race condition while obtaining the Get State Control Primitive values.

Group 8: Management Interface Commands

Test 8.1 – NVMe-MI Response Header – (Mandatory)

Purpose: Reserved field, Integrity Check (verify correct CRC-32)

References:
NVMe-MI Specification Section 4.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 18, 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization
2. Send an NVMe-MI Configuration Get - Transmission Unit Size

Observable Results:
1. Ensure that a successful Response was returned.
2. Ensure that Byte 3, Byte 2, and bits 1, 2 of Byte 1, are reserved.
3. Ensure that the IC bit is set to 1.

Possible Problems: None

Test 8.2 – NVMe-MI Configuration Set – (Mandatory)

Purpose: Configuration Set.
Specifying a reserved identifier in the Configuration Identifier field causes the command to complete with an Invalid Parameter error status.

References:
NVMe-MI Specification Section 5.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization
2. Perform Configuration Set Command with Configuration Identifier 0x03 (MCTP Transmission Unit Size: Size = 64 bytes)
3. Wait for the Response message
4. Perform Configuration Set Command with Configuration Identifier set to 0
5. Wait for the Response message
6. Repeat the above for Configuration Identifier values of 0x56, 0xBF

Observable Results:
1. Ensure that the Set Configuration command returns with successful completion.
2. Ensure that for the rest of the commands:
    a. In the Parameter Error Location field of the response the byte position is set to 8 and bit position - to 0.

Possible Problems: None

Test 8.3 – NVMe-MI Config Get Response – (Mandatory)

Purpose: Responses to Config Get, Health Status Polls, and VPD Read return data in proper format
Configuration Get:
- SMBus/I2C Frequency. NVMe Management Response Bits 23:4 are reserved, bits 3:0 return the frequency encoding.
- MCTP Transmission Unit Size. NVMe Management Response Bits 23:4 are reserved, bits 3:0 return the unit size.
- Specifying a reserved identifier in the Configuration Identifier field causes the command to complete with an Invalid Parameter error status.

References:
NVMe-MI Specification Section 5.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP initialization
2. Perform Configuration Get Command requesting MCTP Transmission Unit Size
3. Wait for the Response message
4. Perform Configuration Get Command requesting SMBus/I2C Frequency
5. Wait for the Response message

Observable Results:
1. Ensure that for both commands proper response is returned, and status is set to Success
2. Ensure that for the Command requesting MCTP Transmission Unit Size:
    a. Bits 23:16 of the NVMe Management Response field are reserved (set to zero)
    b. Bits 15:0 return the default unit size of 0x40
3. Ensure that for the Command requesting SMBus/I2C Frequency Unit Size:
    c. Bits 23:4 of the NVMe Management Response field are reserved (set to zero)
    d. Bits 3:0 return the valid frequency encoding (0 to 3)

Possible Problems: None

Test 8.4 – NVMe-MI Health Status Poll – (Mandatory)

Purpose: The intent of this test is to verify the reserved bits of the Controller Health Status Poll Response are set to zero.

References:
NVMe-MI Specification Section 5.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: February 15 2017

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization
2. Send Configuration Get Health Status Poll
3. Wait for response

Observable Results:
1. Ensure that the proper response is returned, and status is set to Success
2. Ensure that for the following bits are reserved in the response.
    a. Bits 23:0 in DWORD 1
    b. Bits 1:0 and 7:6 in DWORD 2
    c. Bits 31:13 in DWORD 3

Possible Problems: None

Test 8.5 – NVMe-MI Controller Health Status Poll – (M)

Purpose: Controller Health Status Poll:
- The length should correspond to RENT in Management response
- For each Controller Health data structure bytes 15:9 and bits 15:8 in bytes 322 are reserved

References:
NVMe-MI Specification Section 5.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: August 1st, 2024

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: NVMe-MI 1.0 ECN 003 or NVMe-MI 1.0a or higher Not Implemented (M)

Test Procedure:
1. The following procedure is only applicable to a device implementing NVMe-MI 1.0 which has not implemented NVMe.MI 1.0 ECN003.
2. Perform MCTP initialization.
3. Perform Endpoint Discovery.
4. For each Management Endpoint:
    a. Perform Controller Health Status Poll Command with the following parameters: Report all set, Include PCI Functions set, Controller Status Changes set, MAXRENT set to 0x01, all other parameters set to zero. Wait for the Response message

Observable Results:
1. For each Management Endpoint:
    a. Ensure that the proper response is returned, and status is set to Success
    b. Ensure that the following bits are reserved in the response, and cleared to 0
        i. Bits 15:8 in DWORD 1
        ii. Bits 31:24 in DWORD 2
        iii. Bits 31:5 in DWORD 4
        iv. Bits 31:0 in DWORD 5
    c. Ensure that if a Controller Health Data Structure is included in the response, that RENT (bits 23:16 in DWORD 1) uses a 0's based value (i.e a value of 0 indicated 1 Data Structure included).

Case 2: NVMe-MI 1.0 ECN 003 or NVMe-MI 1.0a or higher Implemented (M)

Test Procedure:
1. The following procedure is only applicable to a device implementing NVMe.MI 1.0 which has implemented ECN003 and to devices that implement NVMe-MI 1.0a or higher.
2. Perform MCTP initialization
3. Perform Endpoint Discovery.
4. For each Management Endpoint:
    a. Perform Controller Health Status Poll Command with the following parameters: Report all set, Include PCI Functions set, Controller Status Changes set, MAXRENT set to 0x01, all other parameters set to zero.
    b. Wait for the Response message

Observable Results:

1. For each Management Endpoint:
    a. Ensure that the proper response is returned, and status is set to Success
    b. Ensure that the following bits are reserved in the response, and cleared to 0.
        i. Bits 15:8 in DWORD 1
        ii. Bits 31:24 in DWORD 2
        iii. Bits 31:5 in DWORD 4
        iv. Bits 31:0 in DWORD 5
    c. Ensure that if a Controller Health Data Structure is included in the response, that RENT (bits 23:16 in DWORD 1) is set to a non-zero value

Case 3: Controller Health Status Poll Filtering by Controller Selection (M)

Test Procedure:

1. The following procedure is only applicable to a device implementing NVMe.MI 1.0 which has implemented ECN003 and to devices will implement NVMe-MI 1.0a or higher.
2. Perform MCTP initialization
3. Perform Endpoint Discovery
4. For each Management Endpoint:
    a. Perform Controller Health Status Poll Command with the following parameters:
        i. ALL= 0, INCVF=1, INCPF=0, INCF=0, MAXRENT=255, SCTLID=0
        ii. CWARN=0, SPARE=0, PDLU=0, CTEMP=0
    b. Wait for the Response message
    c. Perform Controller Health Status Poll Command with the following parameters:
        i. ALL= 0, INCVF=0, INCPF=1, INCF=0, MAXRENT=255, SCTLID=0
        ii. CWARN=0, SPARE=0, PDLU=0, CTEMP=0
    d. Wait for the Response message
    e. Perform Controller Health Status Poll Command with the following parameters:
        i. ALL= 0, INCVF=0, INCPF=0, INCF=1, MAXRENT=255, SCTLID=0
        ii. CWARN=0, SPARE=0, PDLU=0, CTEMP=0
    f. Wait for the Response message

Observable Results:

1. For each Management Endpoint:
    a. Verify that for each Controller Health Status Poll Command, only the data structures matching the Controller filtering requirements were returned.
    b. In each case ensure that the proper response is returned, and status is set to Success
    c. Ensure that for the following bits are reserved in the response, and cleared to 0.
        i. Bits 15:8 in DWORD 1
        ii. Bits 31:24 in DWORD 2
        iii. Bits 31:5 in DWORD 4
        iv. Bits 31:0 in DWORD 5
    d. Ensure that if a Controller Health Data Structure is included in the response, that RENT (bits 23:16 in DWORD 1) is set to a non-zero value

Case 4: Controller Health Status Poll Filtering by Error Selection Fields (M)

Test Procedure:

1. The following procedure is only applicable to a device implementing NVMe.MI 1.0 which has implemented ECN003 and to devices will implement NVMe-MI 1.0a or higher.
2. Perform MCTP initialization
3. Perform Endpoint Discovery
4. For each Management Endpoint

a. Perform Controller Health Status Poll Command with the following parameters:
    i. ALL= 0, INCVF=0, INCPF=0, INCF=0, MAXRENT=255, SCTLID=0
    ii. CWARN=1, SPARE=0, PDLU=0, CTEMP=0
b. Wait for the Response message
c. Perform Controller Health Status Poll Command with the following parameters:
    i. ALL= 0, INCVF=0, INCPF=0, INCF=0, MAXRENT=255, SCTLID=0
    ii. CWARN=0, SPARE=1, PDLU=0, CTEMP=0
d. Wait for the Response message
e. Perform Controller Health Status Poll Command with the following parameters:
    i. ALL= 0, INCVF=0, INCPF=0, INCF=0, MAXRENT=255, SCTLID=0
    ii. CWARN=0, SPARE=0, PDLU=1, CTEMP=0
f. Wait for the Response message
g. Perform Controller Health Status Poll Command with the following parameters:
    i. ALL= 0, INCVF=0, INCPF=0, INCF=0, MAXRENT=255, SCTLID=0
    ii. CWARN=0, SPARE=0, PDLU=0, CTEMP=1
h. Wait for the Response message

Observable Results:

1. For each Management Endpoint:
    a. Verify that for each Controller Health Status Poll Command, only the data structures matching the error selection filtering requirements were returned.
    b. In each case ensure that the proper response is returned, and status is set to Success
    c. Ensure that the following bits are reserved in the response, and cleared to 0
        i. Bits 15:8 in DWORD 1
        ii. Bits 31:24 in DWORD 2
        iii. Bits 31:5 in DWORD 4
        iv. Bits 31:0 in DWORD 5
    d. Ensure that if a Controller Health Data Structure is included in the response, that RENT (bits 23:16 in DWORD 1) is set to a non-zero value.

Possible Problems: NVMe-MI 1.0 ECN 003 changed the RENT field from being a zeroes based value to not being a zeroes based value. Products implementing NVMe-MI 1.0 may or may not implement NVMe-MI 1.0 ECN 003. Products implementing NVMe-MI 1.0a or higher must implement NVME-MI 1.0 ECN 003.

Case 5: Controller Health Status Poll Data Verification (FYI)

Test Procedure:

1. Perform MCTP initialization
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform a Controller Health Status Poll command with Report All (ALL Bit 31) set to 0, MAXRENT set to 254, Starting Controller ID (SCTLID) cleared to 0, and the Clear Changed Flags bit set to 1. Record the returned values in the Controller Health Data Structure.
4. Perform a second Controller Health Status Poll command with the same parameters as the previous step except change Clear Changed Flags to be cleared to 0h.
5. Perform a second Controller Health Status Poll command with Report All (ALL Bit 31) set to 0, MAXRENT set to 254, Starting Controller ID (SCTLID) cleared to 0, and the Clear Changed Flags bit set to 1, and CWARN, SPARE, PDLU, CTEMP and CSTS fields all set to 1.

Observable Results:

1. Verify the returned data structure from step 2 contains Namespace Attribute Changed (NAC), Firmware Activated (FA) and Telemetry Controller-Initiated Data Available (TCIDA) and they are all cleared to 0.

2. Verify the returned data structure from step 3 is identical to the data structure returned in step 2.
3. Verify that in the returned Controller Health Data Structure from step 4, that the CWARN, SPARE, PDLU, CTEMP and CSTS fields are reported.
4. Verify that no more than 255 Controller IDs are reported, and that the Controller IDs are returned in ascending order.

Case 6: Controller Health Data Structure Matches SMART/Health Log Page (FYI)

Test Procedure:

1. Perform MCTP initialization
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform a Controller Health Status Poll command with Report All (ALL Bit 31) set to 1, MAXRENT set to 254, Starting Controller ID (SCTLID) cleared to 0, and the Clear Changed Flags bit set to 0. Record the returned values in the Controller Health Data Structure.
4. Perform a Configuration Get with the Configuration Identifier of Health Status Change (02h).
5. Perform a second Controller Health Status Poll command with Report All (ALL Bit 31) set to 0, MAXRENT set to 254, Starting Controller ID (SCTLID) cleared to 0, and the Clear Changed Flags bit set to 1. Record the returned values in the Controller Health Data Structure.
6. Issue a Get Log page with LID 02h (SMART/Health Information) and record the value of the Critical Warning byte

Observable Results:

1. Verify that if any of the following fields were set to 1 in the returned data structure from the first Controller Health Status Poll command, that they are cleared to 0 in the second Controller Health Status Poll command in step 4.
    a. Telemetry Controller-initiated Data Available (TCIDA)
    b. Firmware Activated (FA)
    c. Namespace Attribute Changes (NAC)
    d. Controller Enabled Change Occurred (CECO)
    e. NVM Subsystem Reset Occurred (NSSRO)
2. Verify that the value of the Critical Warning byte from the Get Log Page command is identical to the value returned in the Controller Health Data Structure returned in step 4.

Test 8.6 – NVMe-MI Read Data Structure – (Mandatory)

Purpose:
The intent of this test is to verify the response length and reserved bits of the Get Read Subsystem information response for different Data Structure Types.
References:
NVMe-MI Specification Section 5.7

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: May 27, 2021

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: NVMe-MI Read Data Structure (Mandatory)

Test Procedure:
1. Verify that the device claims support for NVMe-MI v1.1 or previous, if not, this test is not applicable.
2. Perform MCTP initialization
3. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00
4. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x01
5. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x02
6. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x03
7. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x04

Observable Results:
1. Ensure that a proper response is returned with status set to Success
    a. Bits 23:16 of dw0 of the NVMe Management response are reserved for all responses
    b. Check Bits 15:00 of dw0 for response length for all responses
2. Ensure that for the when DTYP is set to Subsystem Information (0x00)
    a. Check that the response length = 12 + 32 = 44 bytes
    b. BYTES 31:03 are reserved
3. Ensure that for the when DTYP is set to Port Information (0x01)
    a. Check that the response length = 12 + 32 = 44 bytes
    b. BYTES 07:04 are reserved
    c. BYTE 9 bits 7:3 are reserved
    d. BYTES 31-13 are reserved
    e. BYTE 01 is reserved
4. Ensure that for the when DTYP is set to Controller List (0x02)
    a. Check that the response length = 12 + 8 = 20 bytes (4 bytes per controller)
5. Ensure that for the when DTYP is set to Controller Information (0x03)
    a. Check that the response length = 12 + 32 = 44 bytes
    b. BYTES 7:1 are reserved
    c. BYTES 4-1 are reserved
    d. BYTE 5 bits 7:1 are reserved
    e. BYTES 31:16 are reserved
6. Ensure that for the when DTYP is set to optional Commands supported (0x04)
    a. Check that the response length = 12 + 6 = 18 bytes (CMD0 and CMD1)
    b. bits 2:0 and bit 7 of BYTE 00 are reserved for each entry in the list
    c. Verify that the PCIe Supported Link Speeds Vector bits are set properly according to the PCIe Link Speeds supported by the DUT .

Possible Problems: None

Case 2: NVMe-MI Read Data Structure, greater than version 1.1 (Mandatory)

Test Procedure:

1. Verify that the device claims support for NVMe-MI v1.1 or beyond, if not, this test is not applicable
2. Perform MCTP initialization
3. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00
4. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x01
5. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x02, CTRLID=Controller Id under test.
6. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x03 CTRLID=Controller Id under test.
7. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x04 CTRLID=Controller Id under test.

Observable Results:

1. Ensure that a proper response is returned with status set to Success.
    a. Bits 23:16 of dw0 of the NVMe Management response are reserved for all responses
    b. Check Bits 15:00 of dw0 for response length for all responses
2. Ensure that for the when DTYP is set to Subsystem Information (0x00)
    a. Check that the response length = 12 + 32 = 44 bytes
    b. BYTES 31:03 are reserved
3. Ensure that for the when DTYP is set to Port Information (0x01)
        i. Bytes are reserved depending on MI spec version conformance
    b. Check that the response length = 12 + 32 = 44 bytes
    c. BYTES 07:04 are reserved
    d. BYTE 9 bits 7:3 are reserved
    e. BYTES 31-13 are reserved
    f. BYTE 01 is reserved
4. Ensure that for the when DTYP is set to Controller List (0x02)
    a. Check that the response length = 12 + 8 = 20 bytes (4 bytes per controller)
5. Ensure that for the when DTYP is set to Controller Information (0x03)
    a. Check that the response length = 12 + 32 = 44 bytes
    b. BYTES 7:1 are reserved
    c. BYTES 4-1 are reserved
    d. BYTE 5 bits 7:1 are reserved
    e. BYTES 31:16 are reserved
6. Ensure that for the when DTYP is set to optional Commands supported (0x04)
    a. Check that the response length = 12 + 6 = 18 bytes (CMD0 and CMD1)
    b. bits 2:0 and bit 7 of BYTE 00 are reserved for each entry in the list
    c. Verify that the PCIe Supported Link Speeds Vector bits are set properly according to the PCIe Link Speeds supported by the DUT .

Case 3: Read NVMe-MI Data Structure NUMCMD ≤ 2047 (FYI)

Test Procedure

1. Configure the NVMe Host to issue a Read NVMe-MI Data Structure command with DTYP = 04h and CTRLID set to the Controller under test.

Observable Results:

1. Verify that the Read NVMe-MI Data Structure command completes successfully.
2. Verify that the returned Data Structure contains no more than 2047 commands.
3. If there are 0 optionally supported commands (i.e. bytes [01:00] are cleared to 0), verify that the remainder of the data structure is zero filled.

Possible Problems: None.

Case 4: Read NVMe-MI Data Structure NMINT≠2h (FYI)

Test Procedure

1. Configure the NVMe Host to issue a Read NVMe-MI Data Structure command with DTYP = 04h and CTRLID set to the specified Controller. Record the value of the NUMCMD field.

Observable Results:

1. Verify that the Read NVMe-MI Data Structure command completes successfully.
2. For each optional command supported in the returned data structure (CMD 0 to CMD NUMCMD-1), verify that the NVMe-MI Message Type (NMIMT) field does not contain a value of 02h.

Test 8.7 – NVMe-MI Data Length – (Mandatory)

Purpose: The intent of this test is to verify the response length of specific requested parameters.

References:
NVMe-MI Specification Section 5.7

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: November 26, 2018

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Case 1: Verify NVMSSI Data Length (M)

Test Procedure:
1. Perform Endpoint Discovery
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00

Observable Results:
1. Verify the response length is 32

Case 2: Verify PortInfo Data Length (M)

Test Procedure:
1. Perform Endpoint Discovery
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x01

Observable Results:
1. Ensure Response Data Length is 32

Case 3: Verify CtrlrList Data Length (M)

Test Procedure:
1. Perform MCTP Initialization
2. Perform Read NVMe-MI Data Structure with DTYP set to 0x02 (Controller List)
3. Wait for Response Message

Observable Results:
1. Ensure that a proper response is returned with status set to Success
    a. Bits 23:16 of dw0 of the response are reserved for all responses
    b. Check Bits 15:00 of dw0 for response length for all responses
    c. Ensure that for the when DTYP is set to Controller List (0x02)
2. Check that the response length = 12 + 8 = 20 bytes (4 bytes per controller)

Case 4: Verify CtrlrInfo Data Length (M)

Test Procedure:
1. Perform MCTP Initialization
2. Perform Read NVMe-MI Data Structure with DTYP set to 0x03 (Controller Information)
3. Wait for Response Message

Observable Results:
1. Ensure that a proper response is returned with status set to Success
2. Bits 23:16 of dw0 of the response are reserved for all responses
3. Check Bits 15:00 of dw0 for response length for all responses
4. Ensure that for the when DTYP is set to Controller Information (0x03)
5. Check that the response length = 12 + 32 = 44 bytes
6. BYTES 4:1 are reserved
7. Bits 7:1 of byte 5
8. BYTES 31:16 are reserved

Case 5: Verify OptCmds Data Length, less than version 1.1 (M)

Test Procedure:
1. Verify that the device claims support for NVMe-MI v1.1 or previous, if not, this test is not applicable.
2. Perform MCTP Initialization
3. Perform Read NVMe-MI Data Structure with DTYP set to 0x04 (Optional Commands)
4. Wait for Response Message

Observable Results:
1. Ensure that a proper response is returned with status set to Success
2. Bits 23:16 of dw0 of the response are reserved for all responses
3. Check Bits 15:00 of dw0 for response length for all responses
4. Ensure that for the when DTYP is set to optional Commands supported (0x04)
5. Check that the response length = 12 + 6 = 18 bytes (CMD0 and CMD1)
6. Bits 2:0 and bit 7 of BYTE 00 are reserved

Possible Problems: None

Case 6: Verify OptCmds Data Length, greater than version 1.1 (FYI)

Test Procedure:
1. Verify that the device claims support for NVMe-MI v1.1 or beyond, if not, this test is not applicable.
2. Perform MCTP Initialization
3. Perform Read NVMe-MI Data Structure with DTYP set to 0x04 (Optional Commands) and CTRLID=Controller Id under test.
4. Wait for Response Message

Observable Results:
1. Ensure that a proper response is returned with status set to Success
2. Bits 23:16 of dw0 of the response are reserved for all responses
3. Check Bits 15:00 of dw0 for response length for all responses
4. Ensure that for the when DTYP is set to optional Commands supported (0x04)
5. Check that the response length = 12 + 6 = 18 bytes (CMD0 and CMD1)
6. Bits 2:0 and bit 7 of BYTE 00 are reserved

Possible Problems: None

Test 8.8 – Management Endpoint Buffer Read – (FYI)

Purpose: The purpose of this test is to verify the proper implementation of the Management Endpoint Buffer Read command.

References:
NVMe-MI Specification Section 5.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 28, 2020

Discussion: The Management Endpoint Buffer Read command allows the Management Controller to read the contents of the Management Endpoint Buffer. This data is returned in the Response Data.

Test Setup: See Appendix A.

Case 1: Management Endpoint Buffer Read DLEN>0 (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Read Command with the following parameters
    a. DOFST=0
    b. DLEN=1

Observable Results:
1. Verify the response length is 1 and indicates status success

Case 2: Management Endpoint Buffer Read DLEN=0 (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Read Command with the following parameters
    a. DOFST=0
    b. DLEN=0

Observable Results:
1. Verify no response data is sent, and response indicates status success

Case 3: DOFST >= Management Endpoint Buffer Size (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.

4. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Read Command with the following parameters
    a. DOFST greater than or equal to the Management Endpoint Buffer Size
    b. DLEN=1

Observable Results:
1. Verify response indicates Invalid Parameter Error
2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates DOFST field

Case 4: (DOFST + DLEN) > Management Endpoint Buffer (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
4. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Read Command with the following parameters
    a. DOFST = Management Endpoint Buffer Size - 1
    b. DLEN=2

Observable Results:
1. Verify response indicates Invalid Parameter Error
2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates DLEN field

Case 5: Management Endpoint Buffer Read after Sanitize Operation (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. If Management Endpoint Buffer Size is not 0h, perform a Sanitize operation to clear the Management Endpoint Buffer.
4. Send a Management Endpoint Buffer Read Command with the following parameters
    a. DOFST = 0
    b. DLEN=1

Observable Results:
1. Verify response indicates Management Endpoint Buffer Cleared Due to Sanitize.

Possible Problems: None

Test 8.9 – Management Endpoint Buffer Write – (FYI)

Purpose: The purpose of this test is to verify the proper implementation of the Management Endpoint Buffer Write command.

References:
NVMe-MI Specification Section 5.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 28, 2020

Discussion: The Management Endpoint Buffer Write command allows the Management Controller to update the contents of the optional Management Endpoint Buffer. The data used to update the Management Endpoint Buffer is transferred in the Request Data included in a Management Endpoint Buffer Write command.

Test Setup: See Appendix A.

Case 1: Management Endpoint Buffer Write DLEN>0 (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Write Command with the following parameters
    a. DOFST=0
    b. DLEN=1

Observable Results:
1. Verify the response indicates status success

Case 2: Management Endpoint Buffer Write DLEN=0 (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Write Command with the following parameters
    a. DOFST=0
    b. DLEN=0

Observable Results:
1. Verify the response indicates status success

Case 3: DOFST > Management Endpoint Buffer Size (FYI)

Test Procedure:

1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
4. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Write Command with the following parameters
    a. DOFST greater than the Management Endpoint Buffer Size
    b. DLEN=1

Observable Results:
1. Verify response indicates Invalid Parameter Error
2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates DOFST field

Case 4: (DOFST + DLEN) > Management Endpoint Buffer (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
4. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Write Command with the following parameters
    a. DOFST = Management Endpoint Buffer Size - 1
    b. DLEN=2

Observable Results:
1. Verify response indicates Invalid Parameter Error
2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates DLEN field

Possible Problems: None

Test 8.10 – SES Read – (FYI)

Purpose: The purpose of this test is to verify the proper implementation of the SES Receive command.

References:
NVMe-MI Specification Section 5.9

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 28, 2020

Discussion: The SES Receive command is used to retrieve SES status type diagnostic pages. Upon successful completion of the SES Receive command, the SES status type diagnostic page is returned in the Response Data. This test is only applicable to NVMe Enclosures.

Test Setup: See Appendix A.

Case 1: SES Receive (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller Data Structure command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Perform a SES Receive command with valid PCODE and ALENGTH fields.

Observable Results:
1. Verify the response indicates status success and includes the requested SES Diagnostic page.

Case 2: SES Receive with Reserved PCODE (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller Data Structure command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
4. Perform a SES Receive command with valid ALENGTH field and a reserved PCODE value.

Observable Results:
1. Verify the response indicates status Invalid Parameter Error.
2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates PCODE field

Case 3: SES Receive with SES Control type PCODE (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
4. Perform a SES Receive command with valid ALENGTH field and a PCODE value that corresponds to a SES Control type diagnostic page.

Observable Results:
1. Verify response indicates Invalid Parameter Error

2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates PCODE field

Case 4: SES Receive with Truncated Diagnostic Page (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Perform a SES Receive command with a valid PCODE value and an ALENGTH value that is less than the expected length of the requested diagnostic page.

Observable Results:
1. Verify response length corresponds to the requested ALENGTH value.

Possible Problems: None

Test 8.11 – SES Send – (FYI)

Purpose: The purpose of this test is to verify the proper implementation of the SES Send command.

References:
NVMe-MI Specification Section 5.10

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 28, 2020

Discussion: The SES Send command is used to transfer SES control type diagnostic pages to an SES Enclosure Service Process. Upon successful completion of the SES Send command, the Request Data, containing an SES control type diagnostic page, is transferred by the Request Message or to the Management Endpoint Buffer.

Test Setup: See Appendix A.

Case 1: SES Send (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller Data Structure command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Perform a SES Send command with a valid SES Diagnostic page and DLEN field.

Observable Results:
1. Verify the response indicates status success.

Case 2: SES Send with DLEN=0 (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller Data Structure command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Perform a SES Send command with DLEN=0.

Observable Results:
1. Verify the response indicates status Success.

Possible Problems: None

Test 8.12 – VPD Read – (M)

Purpose: The purpose of this test is to verify the proper implementation of the VPD Read command.

References:
NVMe-MI Specification Section 5.11

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 28, 2020

Discussion: The VPD Read command is used to read the Vital Product Data described in section 9.2. Upon successful completion of the VPD Read command, the specified portion of the VPD contents is returned in the Response Data.

Test Setup: See Appendix A.

Case 1: VPD Read (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform a VPD Read command with a valid DOFST and non-zero DLEN fields.

Observable Results:
1. Verify the response indicates status success and VPD Response data is returned.

Case 2: VPD Read with DLEN=0 (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform a VPD Read command with a valid DOFST and DLEN=0.

Observable Results:
1. Verify the response indicates status success and no VPD Response data is returned.

Case 3: VPD Read with DLEN + DOFST > VPD Size (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR & MNR version supported.
3. Perform a VPD Read command with a DOFST + DLEN > VPD Size.

Observable Results:
1. Verify that the DUT does not return the VPD contents and responds with an Invalid Parameter Error Response
2. Verify if the DUT supports NVMe-MI version 1.2 or later, then the PEL field indicates DLEN field

Possible Problems: None

Case 3: Management Endpoint Transition To Operational Power State - MUT, MERIMTO, MERWMTO (FYI)

Test Procedure:

1. This test is only applicable to a device supporting NVMe-MI 2.0 or greater. If not, skip this test.
2. Perform Endpoint Discovery.
3. If the NVM Subsystem has a 2-Wire port and the 2-Wire port is in SMBus mode, issue an I2C Read to retrieve the VPD. Otherwise, issue a VPD Read command to retrieve the VPD.
4. Record the values of MUT, MERIMTO, and MERWMTO in the NVM Subsystem Element Descriptor.
5. Configure the NVMe Host to restrict all power for the Management Endpoint.
6. Configure the NVMe Host to bring back power for the Management Endpoint and start a new timer with millisecond granularity.
7. Configure the NVMe Host to issue I2C Read or VPD Read commands (in alignment with what was sent in step 3) until 100*MUT milliseconds have passed on the timer, or until one is processed successfully.
8. Configure the NVMe Host to issue Request Messages that do not require media access until 100*MERIMTO milliseconds have passed on the timer, or until one is processed successfully.
9. Configure the NVMe Host to issue Request Messages that require media access until 100*MERWMTO milliseconds have passed on the timer, or until one is processed successfully.

Observable Results:

1. Verify that MUT, MERIMTO, and MERWMTO are not cleared to 0h.
2. Verify that at least 1 I2C Read or VPD Read command issued in step 7 completed successfully within the 100*MUT millisecond timeout.
3. Verify that at least 1 Request Message issued in step 8 completed successfully within the 100*MERIMTO millisecond timeout.
4. Verify that at least 1 Request Message issued in step 9 completed successfully within the 100*MERWMTO millisecond timeout.

Test 8.13 – VPD Write – (M)

Purpose: The purpose of this test is to verify the proper implementation of the VPD Write command.

References:
NVMe-MI Specification Section 5.11

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 17, 2023

Discussion: The VPD Write command is used to update the Vital Product Data.

Test Setup: See Appendix A.

Case 1: VPD Write with DLEN=0 (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform a VPD Write command with a valid DOFST and DLEN=0.

Observable Results:
1. Verify the response indicates status success.

Case 2: VPD Write Data Check (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform a VPD Write command with a valid DOFST and non-zero DLEN.
3. Perform a VPD Read command to the same offset and length as the previous VPD Write.

Observable Results:
1. Verify the response to each command indicates status success.
2. Verify that the returned VPD Read data matched the data written in the VPD Write.

Case 3: VPD Write Cycle Information Check (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify command to retrieve the Identify Controller Data Structure.

Observable Results:
1. Verify if the VWCRV bit is set to 0, that the VWCR is also set to 0.

Possible Problems: Devices supporting MI 1.2 or beyond, the VPD Write and Reset commands are now optional

Group 9: NVMe Admin Command Set Tests

Test 9.1 – NVMe Identify Command – (M)

Purpose: Check operation mandatory NVMe Admin Commands over NVMe-MI.

References:
NVMe-MI Specification 6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15 ,2021

Discussion: The NVM Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. Certain NVM Express Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device and an NVMe Enclosure using the out-of-band mechanism.

Test Setup: See Appendix A.

Case 1: Identify Command (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Record the NVMSR field.

Observable Results:
1. Verify that a response indicating status success is return along with the Identify Controller data structure.
2. Verify that if the DUT is an NVMe Storage Device, the NVMESD bit is set to 1.
3. Verify that if the DUT is an NVMe Enclosure, the NVMEE bit is set to 1.
4. Verify that the NVMSR field is not cleared to 0.
5. Verify that if the VPD Write Cycle Remaining Valid bit is cleared to ‘0’, then VPD Write Cycles Remaining field is cleared to a value of 0h.

Possible Problems: None

Test 9.2 – NVMe Get Log Command – (FYI)

Purpose: Check operation mandatory NVMe Admin Commands over NVMe-MI.

References:
NVMe-MI Specification 6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: The NVM Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. Certain NVM Express Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device and an NVMe Enclosure using the out-of-band mechanism.

Test Setup: See Appendix A.

Case 1: Get Log Page Command (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMESD is set to 0, this test is not applicable.
3. Perform a Get Log Page command in-band over the NVMe Interface Supported Log Pages.
    a. The following log pages are mandatory
        i. Supported Log Pages
        ii. Error Information
        iii. SMART / Health Information (Controller scope)
        iv. Firmware Slot Information
        v. Feature Identifiers Supported and Effects

Observable Results:
1. Verify that a response indicating status success is returned along with requested Log Page for each supported Log Page.
2. Verify that the mandatory log pages listed in step 3 are supported by the NVMe Management Endpoint

Possible Problems: Supported Log Pages and Feature Identifiers Supported and Effects Log page are optional for devices supporting versions 1.1 and earlier for MI.

Case 2: Get Log Page Command, Retain Asynchronous Event bit cleared (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMSD is set to 0, this test is not applicable.
3. Check the version field. If the DUT supports NVMe base version 1.2 or older, then this test is not applicable
4. Configure the NVMe host to issue Asynchronous Event Requests until there are no outstanding Asynchronous Events. Perform a Controller Level Reset to abort the remaining Asynchronous Event Requests.
5. For each NVMe Interface Supported Log Page that does not deal with Asynchronous Events,
    a. Perform a Get Log Page command in-band with the RAE bit set to 1
    b. Perform a Get Log Page command in-band with the RAE bit cleared to 0.
6. For each NVMe Interface Supported Log Page that deals with Asynchronous Events,

a. Configure the NVMe Host to trigger an Asynchronous Event for the log page.
b. Perform a Get Log Page command in-band with the RAE bit cleared to 0.
c. Configure the NVMe Host to issue an Asynchronous Event Request.

Observable Results:
1. Verify that the Get Log Page commands from step 5 ignore the RAE bit by returning similar status codes for each Log Page Identifier.
2. Verify that the Get Log Page commands from step 6.b complete successfully.
3. Verify that a completion queue for the triggered Asynchronous Event is posted after issuing each Asynchronous Event Request in step 6.c.
4. Verify that all other commands complete successfully.

Case 3: Get Log Page Command, Boot Partition (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Configure the NVMe host to read the CAP.BPS register field, if the value is set to zero this test is not applicable.
3. Perform a Get Log Page command Boot Partition, Log Identifier 15h, with the Boot Partition Identifier equal to the value of CAP.BPS.

Observable Results:
1. Verify that a response indicating the Boot Partition log page, that the Log Identifier is set to 15h and Active Boot Partition ID and Boot Partition Size field are displayed.

Case 4: Get Log Page with DOFST & OT (FYI)

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.2 or greater, if not skip this test Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMSD is set to 0, this test is not applicable.
3. Perform a Get Log Page command in-band over the NVMe Interface to LID 0h with DOFST fields and the OT bit cleared to 0h.
4. Perform a Get Log Page command in-band over the NVMe Interface to LID 0h with DOFST=4h and DLEN=1016 set to and the OT bit set to 1h.

Observable Results:
1. Verify all commands complete with status success.
2. Verify that all the data returned in the second log page matches bytes 4 through 1020 of the first get log page.

Possible Problems: None

Case 5: Management Address List Descriptors (FYI)

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.2 or greater, if not skip this test Perform Endpoint Discovery.
2. Perform a Get Log Page command in band for LID 00h, Supported Log Pages, if LID 18h is not supported then this test is not applicable.
3. Perform a Get Log Page command in-band for LID 18h, Management Address List.

Observable Results:
1. Verify the Get Log Page command for LID 18h completes successfully..
2. Verify if a Management Address Descriptor is found with the Management Address Type (MAT) field set to FFh, all subsequent descriptors set the MAT field to FFh.

Possible Problems: None

Test 9.3 – NVMe Get / Set Feature Command – (FYI)

Purpose: Check operation mandatory NVMe Admin Commands over NVMe-MI.

References:
NVMe-MI Specification 6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: The NVM Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. Certain NVM Express Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device and an NVMe Enclosure using the out-of-band mechanism.

Test Setup: See Appendix A.

Case 1: Get Feature Command (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command in-band over the NVMe Interface to the NVMe Controller to determine all supported Features.
4. Perform a Set Feature command in-band over the NVMe Interface to the NVMe Controller to determine all supported Features.
5. Perform a Get Feature command out-of-band over the NVMe-MI Interface to the Management Endpoint for all supported Features as determined in the earlier steps.
6. Perform a Set Feature command out-of-band over the NVMe-MI Interface to the Management Endpoint for all supported Features as determined in the earlier steps.

Observable Results:
1. Verify that a response indicating status success is returned for each supported Set/Get Feature command.

Case 2: Get Feature Command to Host Metadata FIDs GDHM = 1 (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FIDs 7Dh, 7Eh, 7Fh with the SEL field set to 011b and GDHM set to 1.

Observable Results:
1. Verify that the Saveable bit in Dword 0 is cleared to 0 for the corresponding completion queue entry for each Get Features Commands.
2. Verify that the Changeable bit in Dword 0 is set to 1 for the corresponding completion queue entry for each Get Features Commands.
3. Verify that if the generated vendor specific string's Metadata Element Descriptor does not exist for the Host Metadata Data Structure that contains the default value of the specified Host Metadata Feature value, then the controller shall create the Metadata Element Descriptor in the Host Metadata Data Structure that contains the default value with the generated vendor specific string.
4. Verify that if the generated vendor specific string's Metadata Element Descriptor does exist for the Host Metadata Data Structure that contains the default value of the specified Host Metadata Feature value, then the controller shall replace the Metadata Element Descriptor with the generated vendor specific string.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosure devices

Case 3: Get Feature Command to Host Metadata FIDs GDHM = 0 (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform an NVMe Subsystem Reset on all Management Endpoints.
4. Perform a Get Feature command for FIDs 7Dh, 7Eh, 7Fh with the SEL field set to 011b and GDHM set to 0.

Observable Results:
1. Verify that the Saveable bit in Dword 0 is cleared to 0 for the corresponding completion queue entry for each Get Features Commands.
2. Verify that the Changeable bit in Dword 0 is set to 1 for the corresponding completion queue entry for each Get Features Commands.
3. Verify that the device does not generate any vendor specific strings for the Element Types of the specified Host Metadata feature.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosure devices

Case 4: Set Feature Command to EA = 00b Non Existent Element Type (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FIDs 7Eh, 7Fh each with the GDHM field set to 1b
4. Perform a Set Feature command for FIDs 7Eh, 7Fh each with the EA field set to 00b and the Element Type set to a value that does not exist in the specified Host Metadata Feature value. The Get Feature responses can be used to determine a non-existent Element Type.

Observable Results:
1. Verify that the Controller creates the descriptor in the specified Host Metadata Feature value with the value in the Host Metadata data structure.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosure devices

Case 5: Set Feature Command to EA = 00b Element Type Exists (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FIDs 7Eh, 7Fh each with the GDHM field set to 1b
4. Perform a Set Feature command for FIDs 7Eh, 7Fh each with the EA field set to 00b and the Element Type set to a value that does not exist in the specified Host Metadata Feature value. The DLEN of the Set Feature should be set to a value greater than 4096 that does not have bits 1:0 set to a non-zero value. The Get Feature responses can be used to determine a non-existent Element Type.

Observable Results:

1. Verify that the Set Feature causes the Controller to respond with a status of 'Invalid Parameter Error Response' with the PEL field indicating the DLEN field.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosures devices

Case 6: Set Feature Command to FID Enhanced Controller Metadata 7Dh, EA = 00b (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Set Feature command for FID 7Dh and the EA field set to 00b.

Observable Results:

1. Verify that the Controller aborts the Set Features command with status 'Invalid Field in Command'.

Case 7: Set Feature Command to FID Enhanced Controller Metadata 7Dh, EA = 01b (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Dh, 7Eh, 7Fh and the GDHM = 1.
4. Perform a Set Feature command for FID 7Dh, 7Eh, 7Fh and the EA field set to 01b for existing Metadata Element Descriptors.
5. Perform a Get Feature command for FID 7Dh, 7Eh, 7Fh and the GDHM = 1.
6. Perform a Set Feature command for FID 7Dh, 7Eh, 7Fh and the EA field set to 01b for non-existent Metadata Element Descriptors.
7. Perform a Get Feature command for FID 7Dh, 7Eh, 7Fh and the GDHM = 1.

Observable Results:

1. Verify that the Set Feature command completes with status success and the controller deletes all the specified Metadata Element descriptors when performed for existing Metadata Element Descriptors.
2. Verify that the Set Feature command completes with status success and the controller does not change or update any existing Metadata Element Descriptors.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosure devices

Case 8: Set Feature Command to FID Enhanced Controller Metadata 7Dh, EA = 10b, Descriptor does not exist (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Dh and the GDHM = 1.
4. Perform a Set Feature command for FID 7Dh and the EA field set to 10b for Metadata Element Descriptors that do not exist.
5. Perform a Get Feature command for FID 7Dh and the GDHM = 1.

Observable Results:
1. Verify that the Set Feature command completes with status success and the controller created new Metadata Element Descriptors which were returned in response to the second Get Feature command.

Case 9: Set Feature Command to FID Enhanced Controller Metadata 7Dh, EA = 10b, Descriptor Exists (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Dh and the GDHM = 1.
4. Perform a Set Feature command for FID 7Dh and the EA field set to 10b for Metadata Element Descriptors that do exist.
5. Perform a Get Feature command for FID 7Dh and the GDHM = 1.

Observable Results:
1. Verify that the Set Feature command completes with status success and the controller added the specified Metadata Element to the Enhanced Controller Metadata Feature value and did not modify any existing Metadata Element Descriptors.

Case 10: Set Feature Command to FID 7Eh and 7FH, EA = 10b (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Set Feature command for FID 7Eh and the EA field set to 10b.
4. Perform a Set Feature command for FID 7Fh and the EA field set to 10b.

Observable Results:
1. Verify that each Set Feature command was aborted with status 'Invalid Field in Command'.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosure devices

Case 11: Host Metadata Feature too Large (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Dh and the GDHM = 1.
4. Perform a Set Feature command for FID 7Dh and the EA field set to 10b for Metadata Element Descriptors that do exist.
5. Perform a Get Feature command for FID 7Dh and the GDHM = 1.

6. Repeat steps 4 and 5 until the Host Metadata Feature value exceeds 4 KiB.

Observable Results:
1. Verify that the Set Feature command which causes the Host Metadata Feature value to exceed 4 KiB completes with status status 'Invalid Field in Command'.

Case 12: Enhanced Controller Metadata Feature value after Reset (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Dh and the GDHM = 1.
4. Perform a Controller Level Reset.
5. Perform a Get Feature command for FID 7Dh and the GDHM = 1.

Observable Results:
1. Verify that the value for the Number of Metadata Element Descriptors of the Enhanced Controller Metadata Feature returned in response ot the second get Feature command is set to 0h.

Case 13: Get Feature Command to FID 7Dh, SEL= 011b (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Dh and the GDHM = 1, SEL=011b.

Observable Results:
1. Verify that the NS Specific bit in Dword 0 of the corresponding completion queue entry for the Get Feature command is cleared to '0'.

Case 14: Get Feature Command to FID 7Fh, SEL= 011b (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FID 7Fh and SEL=011b.

Observable Results:
1. Verify that the NS Specific bit in Dword 0 of the corresponding completion queue entry for the Get Feature command is set to '1'.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosure devices

Case 15: Set Feature Command DLEN > 4096 (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. Perform a Get Feature command for FIDs 7Eh, 7Fh each with the GDHM field set to 1b
4. Perform a Set Feature command for FIDs 7Eh, 7Fh each with the EA field set to 00b and the Element Type set to a value that does not exist in the specified Host Metadata Feature value. The DLEN of the Set Feature should be set to a value greater than 4096. The Get Feature responses can be used to determine a non-existent Element Type.

Observable Results:

1. Verify that the Set Feature causes the Controller to respond with a status of 'Invalid Parameter Error Response' with the PEL field indicating the DLEN field.

Possible Problems: FID 7Fh is optional in MI 1.2 NVMe Enclosures devices

Case 16: Prohibited Features for Management Endpoint: NVMe Storage Device (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMESD is set to 0, this test is not applicable.
3. Perform a Set Feature command out-of-band over the NVMe-MI Interface to the Management Endpoint for all of the following prohibited features:
    1. Arbitration
    2. LBA Range Type
    3. Error Recovery
    4. Volatile Write Cache
    5. Number of Queues
    6. Interrupt Coalescing
    7. Interrupt Vector Configuration
    8. Write Atomicity Normal
    9. Asynchronous Event Configuration
    10. Host Memory Buffer
    11. Keep Alive Timer
    12. Read Recovery Level Config
    13. Predictable Latency Mode Config
    14. Predictable Latency Mode Window
    15. LBA Status Information Attributes
    16. Host Behavior Support
    17. Software Progress Marker
    18. Host Identifier
    19. Reservation Notification Mask
    20. Reservation Persistence
    21. Namespace Write Protection Config

Observable Results:

1. Verify that all of the Set Feature commands performed in step 3 completed with an unsuccessful status.

Case 17: Embedded Management Controller Address (78h) Set and Get Feature (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform a Get Features command with Feature Identifier (FID) field set to 78h and the Select (SEL) field set to 000b. If the feature is indicated as not changeable by the controller then this test is not applicable.
3. Perform a Set Features command with Feature Identifier (FID) field set to 78h and the Select (SEL) field set to 000b, and a new valid URI in the Embedded Management Controller Address (EMCA) field.
4. Perform a second Get Features command.

Observable Results:

1. Verify the Get and Set Features commands complete successfully.

2. Verify the data returned in the second Get Features command matches the values set by the Set Features command in step 3

Case 17: Host Management Agent Address (79h) Set and Get Features (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform a Get Features command with Feature Identifier (FID) field set to 79h and the Select (SEL) field set to 000b. If the feature is indicated as not changeable by the controller, then this test is not applicable.
3. Perform a Set Features command with Feature Identifier (FID) field set to 79h and the Select (SEL) field set to 000b, and a new valid URI in the Host Management Agent Address (HMAA) field.
4. Perform a second Get Features command.

Observable Results:

1. Verify both the Get and Set Features commands complete successfully.
2. Verify the data returned in the second Get Features command matches the values set by the Set Features command in step 3.

Case 17: Set Feature Command DLEN field bits 1:0 set to a Non-Zero Value (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Perform an Identify Controller Command
3. Perform a Get Feature command for FIDs 7Eh, 7Fh each with the GDHM field set to 1b
4. Perform a Set Feature command for FIDs 7Eh, 7Fh each with the EA field set to 00b and the Element Type set to a value that does not exist in the specified Host Metadata Feature value. The DLEN of the Set Feature should be set to a value with bits 1:0 set to a non-zero value. The Get Feature responses can be used to determine a non-existent Element Type.

Observable Results:

1. Verify that the Set Feature causes the Controller to respond with a status of 'Invalid Parameter Error Response' with the PEL field indicating the DLEN field.

Test 9.4 – Admin Commands Prohibited Out of Band – (FYI)

Purpose: Check that NVMe Admin commands issued using the out of band mechanism are handled properly .

References:
NVMe-MI Specification 6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 17, 2023

Discussion: The NVM Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. NVM Express Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device or NVMe Enclosure using the out-of-band mechanism are defined in the NVMe-MI specification.

Test Setup: See Appendix A.

Case 1: Admin Commands Prohibited out of Band – Storage Device (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMSD is set to 0, this test is not applicable.
3. Perform the following commands using an out of band mechanism (the NVMe-MI transport)
    a. Abort Command
    b. Asynchronous Event Request
    c. Create I/O Completion Queue
    d. Create I/O Submission Queue
    e. Delete I/O Completion Queue
    f. Delete I/O Submission Queue
    g. Keep Alive
    h. NVMe-MI Send
    i. NVMe-MI Receive

Observable Results:
1. Verify that if the DUT supports NVMe-MI 1.1, a response indicating status Invalid Parameter Error Response with Parameter Error Location pointing to the NVMe opcode is returned for each prohibited command.
2. Verify that if the DUT supports NVMe-MI 1.2 or later, a response indicating status Invalid Parameter Error or Invalid Command Opcode Response with Parameter Error Location pointing to the NVMe opcode is returned for each prohibited command.

Case 2: Admin Commands Prohibited out of Band – Enclosure (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command. Check the NVMSR field. If NVMEE is set to 0, this test is not applicable.
3. Perform the following commands using an out of band mechanism (the NVMe-MI transport)
    a. Abort Command
    b. Asynchronous Event Request
    c. Create I/O Completion Queue

d. Create I/O Submission Queue
e. Delete I/O Completion Queue
f. Delete I/O Submission Queue
g. Keep Alive
h. NVMe-MI Send
i. NVMe-MI Receive
j. Format NVM

Observable Results:

1. Verify that if the DUT supports NVMe-MI 1.1, a response indicating status Invalid Parameter Error Response with Parameter Error Location pointing to the NVMe opcode is returned for each prohibited command.
2. Verify that if the DUT supports NVMe-MI 1.2 or later, a response indicating status Invalid Parameter Error or Invalid Command Opcode Response with Parameter Error Location pointing to the NVMe opcode is returned for each prohibited command.

Possible Problems: Devices supporting version 1.2 or beyond, the NVMe-MI Received command for Read NVMe-MI Data Structure and Configuration Get, along with NVMe-MI Send for Configuration Set, VPD Write and Reset are optional.

Test 9.5 – Sanitize Command – (M)

Purpose: Check operation of NVMe-MI commands during a the Sanitize operation.

References:
NVMe-MI Specification 6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 28, 2020

Discussion: The NVM Express Admin Command Set allows NVMe Admin Commands to be issued to any Controller in the NVM Subsystem using the out-of-band mechanism. Certain NVM Express Admin Commands that are mandatory, optional, and prohibited for an NVMe Storage Device and an NVMe Enclosure using the out-of-band mechanism.

Test Setup: See Appendix A.

Case 1: NVMe-MI Commands During Sanitize (M)

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an Identify Controller command.
3. On the NVMe Interface, initiate a Sanitize operation using supported parameters.
4. While the Sanitize Operation is being performed, perform each of the following Management Interface commands (if supported) on the NVMe-MI Interface:
    a. Configuration Get
    b. Configuration Set
    c. Controller Health Status Poll
    d. Management Endpoint Buffer Read
    e. Management Endpoint Buffer Write
    f. NVM Subsystem Health Status Poll
    g. Read NVMe-MI Data Structure
    h. Reset
    i. SES Receive
    j. SES Send
    k. VPD Read
    l. VPD Write

Observable Results:
1. Verify that each of the NVMe-MI commands supported by the DUT completed successfully.

Possible Problems: Devices supporting MI 1.2 or beyond, the VPD Write and Reset commands are now optional

Test 9.6 – Format NVM, More Processing Time Required – (FYI)

Purpose: To perform a Format NVM command, possibly get a More Processing required Time Response.

References:
NVMe-MI Specification 4.1.2.3 & 4.2.2.1

Resource Requirements:

Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: December 8, 2022

Discussion: A Management Endpoint should only return a More Processing Required Response for a Command Message that are expected to take longer than the required time (e.g. Format NVM).

Test Setup: See Appendix A.

Case 1: NVMe-MI Format NVM, More Processing Required (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Perform an Identify Controller command.
3. On the NVMe Interface, initiate a NVM Format operation using supported parameters.

Observable Results:
1. Verify that the Format NVM command completed successfully.
2. If a More Processing Required Time Response is received wait the amount of time in the More Processing Required Time (MPRT) field unless it is unusually large. Verify that the Success Response has the RR bit set to 1.
3. Verify that only 2 “More Processing Required Response” for a given Command Message instance are sent

Possible Problems: None

Test 9.7 – Admin Command Request Data Offset Exceeds Data Expected Completion Data– (FYI)

Purpose: To perform verify a management endpoint which supports a management endpoint buffer returns a return a Invalid Parameter Error Response when an Admin command with the DOFST is larger than amount of data that is expected to be returned by the command completion.

References:
NVMe-MI Specification 2.0, Section 6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: July 31 2025

Test Setup: See Appendix A.

Case 1: Supported Admin Commands (FYI)

Test Procedure:
1. Perform Endpoint Discovery
2. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
3. Configure the host to issue an Identify Controller Command to the DUT. Check the returned Identify Controller data structure for any Admin Commands that are supported.
4. For every NVMe Admin Command the DUT supports, configure the host to issue an NVMe Admin Request Command with the Data Offset (DOFST) field set to a value greater than expected completion data of the sent command and less than the Memory Endpoint Buffer.

Observable Results:
1. Verify that all NVMe Admin Commands issued in step 3 return with an Invalid Parameter Error Response with the PEL field indicating the DOFST field that was used in the sent NVMe Admin Request Command

Group 10: Management Enhancement Tests

Test 10.1 – NVMe-MI Identify Structure ME Capabilities – (In Progress)

Purpose: New Identify Controller structure fields (ME Capabilities)

References:
TBD

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
TBD

Observable Results:
TBD

Possible Problems: None

Test 10.2 – NVMe-MI Identify Capabilities – (In Progress)

Purpose: In NVMe Identify Controller data structure bytes 254:240 and byte 255 bits 7:2 are reserved. Bits 0 and 1 in byte 255 identify presence of the Management Endpoint on the port (SMBus/I2c and PCIe) – should probably use this information from the start to decide what can be tested for the port.

References:
TBD

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: No further discussion available at time of writing

Test Setup: See Appendix A.

Test Procedure:
TBD

Observable Results:
TBD

Possible Problems: None

Test 10.3 – NVMe-MI Namespace Metadata – (FYI)

Purpose: Verify proper handling of Namespace Metadata Elements.

References:
NVMe-MI Specification Section 8.2.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 24, 2020

Discussion: This feature is used to store metadata about a namespace associated with a Controller in the NVM Subsystem for later retrieval. The values stored in the Namespace Metadata Feature do not modify Controller behavior on the namespace. This feature is namespace specific.

If a Set Features command is submitted for this Feature, a Host Metadata data structure is transferred in the data buffer for the command. The Host Metadata data structure is 4 KiB in size and contains zero or more Metadata Element Descriptors. If host software attempts to add or update a Metadata Element Descriptor that causes the stored Host Metadata data structure to grow larger than 4 KiB, the Controller shall abort the command with an Invalid Parameter Error Response. The Host Metadata structure for this feature is independent of the Host Metadata data structure for the Controller Metadata feature.

Test Setup: See Appendix A.

Case 1: Perform Set/Get Feature for Namespace Metadata (M)

Test Procedure:

1. Perform MCTP initialization.
2. Perform a Get Feature Operation for Feature Value 7Fh Namespace Metadata. Wait for the Get Feature Response.
3. Perform a Set Feature Operation for Feature Value 7Fh Namespace Metadata providing a new supported value. Wait for the Get Feature Response.
4. Perform a Get Feature Operation for Feature Value 7Fh Namespace Metadata. Wait for the Get Feature Response.

Observable Results:

1. Verify that for each request the proper response is returned, and status is set to Success.
2. Verify that the new values provided in the Set Feature operation are returned in the Get Feature operation.

Case 2: Host Namespace Metadata Data Structure Too Large (M)

Test Procedure:

1. Perform MCTP initialization.
2. Perform a Get Feature Operation for Feature Value 7Fh Namespace Metadata. Wait for the Get Feature Response.
3. Perform a Set Feature Operation for Feature Value 7Fh Namespace Metadata that adds or update a Metadata Element that causes the stored Host Metadata data structure to grow larger than 4096 bytes.
4. Perform a Get Feature Operation for Feature Value 7Fh Namespace Metadata. Wait for the Get Feature Response.

Observable Results:

1. Verify that for the initial Get Feature request the proper response is returned, and status is set to Success.
2. Verify that for the Set Feature request the proper response is returned, and status is set to Invalid Parameter.
3. Verify that the new values provided in the Set Feature operation are not returned in the Get Feature operation.

Possible Problems: None

Test 10.4 – NVMe-MI Controller Metadata – (FYI)

Purpose: Verify proper handling of Controller Metadata Elements.

References:
NVMe-MI Specification Section 8.2.1

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 20 2016

Discussion: This feature is used to store metadata about the host platform in an NVM Subsystem for later retrieval. The values stored in the Controller Metadata Feature do not modify Controller behavior.

If a Set Features command is submitted for this Feature, a Host Metadata data structure is transferred in the data buffer for the command. The Host Metadata data structure is 4 KiB in size and contains zero or more Metadata Element Descriptors. If host software attempts to add or update a Metadata Element that causes the stored Host Metadata data structure to grow larger than 4 KiB, the Controller shall abort the command with an Invalid Parameter Error Response. The Host Metadata Data Structure for this feature is independent of the Host Metadata data structure for the Namespace Metadata feature

Test Setup: See Appendix A.

Case 1: Perform Set/Get Feature for Controller Metadata (M)

Test Procedure:
1. Perform MCTP initialization.
2. Perform a Get Feature Operation for Feature Value 7EFh Controller Metadata. Wait for the Get Feature Response.
3. Perform a Set Feature Operation for Feature Value 7Eh Controller Metadata providing a new supported value. Wait for the Get Feature Response.
4. Perform a Get Feature Operation for Feature Value 7Eh Controller Metadata. Wait for the Get Feature Response.

Observable Results:
1. Verify that for each request the proper response is returned, and status is set to Success.
2. Verify that the new values provided in the Set Feature operation are returned in the Get Feature operation.

Case 2: Host Controller Metadata Data Structure Too Large (M)

Test Procedure:
1. Perform MCTP initialization.
2. Perform a Get Feature Operation for Feature Value 7Eh Controller Metadata. Wait for the Get Feature Response.
3. Perform a Set Feature Operation for Feature Value 7Eh Controller Metadata that adds or update a Metadata Element that causes the stored Host Metadata data structure to grow larger than 4096 bytes.
4. Perform a Get Feature Operation for Feature Value 7Eh Controller Metadata. Wait for the Get Feature Response.

Observable Results:
1. Verify that for the initial Get Feature request the proper response is returned, and status is set to Success.
2. Verify that for the Set Feature request the proper response is returned, and status is set to Invalid Parameter.

3. Verify that the new values provided in the Set Feature operation are not returned in the Get Feature operation.

Possible Problems: None**.**

Group 11: Vital Product Data Tests

Test 11.1 – VPD Read Default Values – (FYI)

Purpose: Verify that the DUT sets NVMe-MI VPD default values properly.

References:
NVMe-MI Specification Section 9.2.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 23, 2020

Discussion: The optional Product Info Area shall have the same format and conventions as the Product Info Area Format as defined by the IPMI Platform Management FRU Information Storage Definition. Therefore, all fields within the Product Info Area shall not follow the conventions defined in section 1.8 of the NVMe-MI specificaiton. The Product Info Area factory default values shall be set to the values defined below.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform VPD Read
4. Wait for Response

Observable Results:
1. Ensure that VPD Read returned with Success.
5. Ensure that the Following values were reserved in the Response.
    a. Common Header:
        i. Byte 01 = 0x01 (IPMIVER)
        ii. Byte 04 = 0x01 (PIAOFF)
        iii. Byte 05 = 0x0F (MRIOFF)
        iv. Byte 06 = 0x00 (Reserved)
    b. Product Info Area:
        i. IPMIVER = 01h
        ii. LCODE = 19h
        1. EOR = C1h
        iii. CPIA field is preceded by a Type/Length field
        iv. Padding of 0h to make the area size a multiple of 8 bytes.

Possible Problems: None.

Test 11.2 – Topology Multirecord Area – (M)

Purpose: Verify that the DUT sets Topology Multirecord values properly.

References:
NVMe-MI Specification Section 9.2.5

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 23, 2020

Discussion: This MultiRecord describes an NVMe Storage Device's architectural elements and their connections. It is required on all NVMe Storage Device FRUs.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform VPD Read requesting the Topology Multirecord Area.
4. Wait for Response

Observable Results:
1. Verify that that VPD Read returned with Success.
2. Verify that Topology Record Type ID = 0Dh
3. Verify that Record Format = 02h, or 82h for the last record in list.
4. Verify that the RLEN field is correct.
5. Verify that Record and Header Checksum are correct
6. Verify that Version Number = 0h
7. Verify that Element Count is correct and not set to 0.

Possible Problems: None

Test 11.3 – NVMe Multirecord Area – (M)

Purpose: Verify that the DUT sets NVMe Multirecord values properly.

References:
NVMe-MI Specification Section 9.2.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 23, 2020

Discussion: The NVMe MultiRecord is used to describe the form factor, power requirements, and capacity of NVMe Storage Devices with a single NVM Subsystem.

Test Setup: See Appendix A.

Case 1: NVMe Multirecord Area (M)

Test Procedure:
1. Perform MCTP Initialization.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform VPD Read requesting the NVMe Multirecord Area.
4. Wait for Response

Observable Results:
1. Verify that that VPD Read returned with Success.
2. Verify that NVMe Record Type ID = 0Bh
3. Verify that Record Format = 02h, or 82h for the last record in list.
4. Verify that the RLEN field = 20h or 3Bh
5. Verify that Record and Header Checksum are correct
6. Verify that NVMe Multirecord Area Version Number = 0h

Possible Problems: None

Case 2: Management Endpoint, MultiRecord Info Area Fields (M)

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 2.0 or greater. If not, skip this test.
2. Perform Endpoint Discovery
3. If the NVM Subsystem has a 2-Wire port and the 2-Wire port is in SMBus mode, issue an I2C Read to retrieve the VPD. Otherwise, issue a VPD Read command to retrieve the VPD.
4. Read bytes 7:0 of the VPD Elements to determine the MultiRecord Info Area Starting Offset (MRIOFF) value.
5. Read the contents of the MultiRecord Info Area and record the element descriptors.

Observable Results:
1. Verify that byte 01 of the NVM Subsystem Element Descriptor is set to 1h.
2. Verify that Management Endpoint Ready Independent of Media Timeout (MERIM) is less than or equal to 100.
3. Verify that Management Endpoint Ready With Media Timeout (MERWMTO) is less than or equal to 100.
4. Verify that Maximum Unresponsive Time (MUT) is not cleared to 0h.
5. Verify that the Revision Byte offset 01 is set to 1h in the FRU Information Device Element Descriptor.

6. Verify that the Boot Failure Code Support bit in the FRU Information Device Element Descriptor is not cleared to 0b.
7. Verify that all commands complete successfully.

Test 11.4 – NVMe PCIe Port Area – (M)

Purpose: Verify that the DUT sets NVMe PCIe Port Multirecord values properly.

References:
NVMe-MI Specification Section 9.2.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: April 23, 2020

Discussion: The NVMe PCIe Port MultiRecord is used to describe the PCIe connectivity for NVMe Storage Devices with a single NVM Subsystem.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform VPD Read requesting the NVMe PCIe Port Multirecord Area.
4. Wait for Response

Observable Results:
1. Verify that that VPD Read returned with Success.
2. Verify that NVMe Record Type ID = 0Ch
3. Verify that Record Format = 02h, or 82h for the last record in list.
4. Verify that the RLEN field = 08h or 0Bh
5. Verify that Record and Header Checksum are correct
6. Verify that NVMe PCIe Port Multirecord Area Version Number = 1h

Possible Problems: The NVMe PCIe Port MultiRecord Area should be supported in NVMe-MI version 1.1 and later.

Test 11.5 – FRU Information Device Read via VPD Read – (M)

Purpose: Verify that the VPD properly contains the FRU Information Device.

References:
NVMe-MI Specification Section 9.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: The Vital Product Data (VPD) is FRU Information (refer to the IPMI Platform Management FRU Information Storage Specification) describing an NVMe Storage Device. Each NVMe Storage Device FRU shall have a FRU Information Device with a size of 256 to 4096 bytes which contains the VPD.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform VPD Read requesting the FRU Information Device.
4. Wait for Response

Observable Results:
1. Verify that the VPD Read returned with Success and the FRU Information Device is between 256 to 4096 bytes.
2. Verify if the DUT contains an SMBus/I2C Port, that the FRU Information Device Element Descriptor contains a valid SMBus/I2C Address Info field.
3. Verify if the DUT does not contain an SMBus/I2C Port, that the FRU Information Device Element Descriptor SMBus/I2C Address Info field is cleared to 0h.

Possible Problems: None

Test 11.6 – FRU Information Device Read via I2C Read – (FYI)

Purpose: Verify that the VPD properly contains the FRU Information Device.

References:
NVMe-MI Specification Section 9.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: The Vital Product Data (VPD) is FRU Information (refer to the IPMI Platform Management FRU Information Storage Specification) describing an NVMe Storage Device. Each NVMe Storage Device FRU shall have a FRU Information Device with a size of 256 to 4096 bytes which contains the VPD.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform I2C Read requesting the FRU Information Device.
4. Wait for Response

Observable Results:
1. Verify that the I2C Read returned with Success and the FRU Information Device is between 256 to 4096 bytes.
2. Verify that the FRU Information Device Element Descriptor contains a valid SMBus/I2C Address field.

Possible Problems: None

Test 11.7 – FRU Information Device Update – (FYI)

Purpose: Verify that the VPD can properly update the FRU Information Device.

References:
NVMe-MI Specification Section 9.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: Updating the VPD by writing to the FRU Information Device using I2C Writes shall not be supported if the VPD Write command is supported.

Test Setup: See Appendix A.

Test Procedure:
1. Determine if the DUT supports the VPD Write command. If the DUT does not support the VPD Write command this test is not applicable.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform VPD Read requesting the FRU Information Device.
4. Wait for Response
5. Perform VPD Write updating the FRU Information Device.
6. Perform VPD Read requesting the FRU Information Device to verify that the update was successful.
7. Perform an I2C/SMBus Write to update the FRU Information Device.

Observable Results:
1. Verify that the VPD Write operation was successful.
2. Verify that the I2C/SMBus Write to update the FRU Information Device was not successful.

Possible Problems: Devices supporting MI 1.2 or beyond, the VPD Write and Reset commands are now optional

Test 11.8 – FRU Information Device Internal Offset – (FYI)

Purpose: Verify that the DUT properly contains the FRU Information Device internal offset.

References:
NVMe-MI Specification Section 9.2

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: If an I2C Read is issued, then data is returned from the internal offset within the FRU Information Device and then the internal offset is incremented by 1h. If the Management Controller reads the last byte of the FRU Information Device (refer to Maximum FRU Information Size) via an I2C Read, then the internal offset shall be cleared to 0h (i.e., rolls over to 0h). If only one byte of the Command Offset is provided by the Management Controller, then the least significant byte of the internal offset shall be set to that value and the most significant byte of the internal offset shall be cleared to 0h.

The internal offset shall be cleared to 0h following a power cycle of the FRU Information Device. Implementations are allowed to maintain the current internal offset value or clear it to 0h following a reset of the FRU Information Device.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP Initialization.
2. Verify that the device claims support for NVMe-MI v1.1 and beyond, if not, this test is not applicable.
3. Perform I2C Read requesting the FRU Information Device.
4. Wait for Response
5. Perform I2C Read requesting the FRU Information Device.
6. Wait for Response
7. Continue performing I2C reads until the last byte of the FRU Information Device is read. Perform 1 additional I2C read.
8. Power cycle the device, and allow it to restart and return to the Idle state.
9. Perform I2C Read requesting the FRU Information Device.
10. Wait for Response

Observable Results:
1. Verify that each I2C Read returned with Success and after each Read the internal offset was incremented by 1h.
2. Verify that the internal offset rolls over to 0h when the last FRU Information Device byte is read.
3. Verify that the I2C read after the power cycle returned data from offset 0h.

Possible Problems: None

Test 11.9 – 2-Wire Port Access– (FYI)

Purpose: Verify that the DUT properly ignores VPD/Mux Access when it is not advertised as supported by the management endpoint.

References:
NVMe-Base Specification Revision 2.1: 5.1.13.2.1 I

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 23, 2025

Test Setup: See Appendix A.

Case 1: 2-Wire Port VPD/Mux Access NACKed When Not Supported (FYI)

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 2.0 or greater. If not, skip this test.
2. Configure the NVMe Host to issue an Identify command to retrieve the Identify Controller (CNS 06) Data Structure. Record the value of FAWR.
3. Perform Endpoint Discovery.
4. If the NVM Subsystem has a 2-Wire port and the 2-Wire port is in SMBus mode, and FAWR is cleared to 0b, then this test is not applicable. If the bit is set to 1b and the NVM subsystem has a 2-Wire port, configure the NVMe Host to issue a Firmware Commit command with Commit Action (CA) = 011b to trigger a firmware activation immediately without reset.
5. If the NVMe Subsystem is not a 2-Wire port in SMBus mode, configure the NVMe Host to issue a VPD read command.
6. While the previously issued command is still being processed, configure the Management Controller to perform an 2-Wire Port VPD access or 2-Wire Port Mux Access.

Observable Results:
1. Verify that the Read issued in step 6 is NACKed and no response is received.
2. Verify that all other commands complete successfully.

Possible Problems: None

Case 2: 2-Wire Port Operations Unaffected (FYI)

Test Procedure:
1. Perform Endpoint Discovery.
2. Configure the NVMe Host to clear the CC.EN register field of the NVMe Controller to ‘0’ to disable the controller.
3. While the controller is disabled (CC.EN bit cleared to 0), configure the NVMe Host to perform the following operations if supported:
    a. 2-Wire Port VPD Access
    b. 2-Wire Port Mux Access
    c. 2-Wire Port MCTP Access
    d. PCIe MCTP Access
4. Configure the NVMe Host to poll the CSTS.RDY register field of the NVMe Controller until it transitions to 0.
5. Configure the NVMe Host to set the CC.EN register field of the NVMe Controller to ‘1’ to re–enable the controller (be sure to set required registers field prior to re-enabling the controller).

Observable Results:

1. Verify that all supported commands complete successfully.

Group 12: Management Endpoint Reset Tests

Test 12.1 – NVMe-MI PCIe Endpoint Reset – (FYI)

Purpose: Verify that Management Endpoint resets are properly isolated.

References:

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: May 27, 2021

Discussion: A reset of a Management Endpoint in an NVM Subsystem shall not affect any other Management Endpoint in the NVM Subsystem or any other NVM Subsystem entity. Note that for implementations compliant to NVMe-MI version 1.1 and earlier, during a PCI Express conventional reset of a PCIe Management Endpoint, MCTP accesses may not be supported on other PCIe or 2 Wire Management Endpoints in the NVM Subsystem.

This test is only applicable to products with multiple Management Endpoints within the NVM Subsystem.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP accesses to all management endpoints in the NVM Subsystem.
2. Perform a PCIe Conventional Reset on a single PCIe port

Observable Results:
1. Verify that MCTP accesses on any management endpoints not associated with the reset PCIe port completed without error.

Possible Problems: None

Test 12.2 – NVMe-MI SMBus Endpoint Reset – (FYI)

Purpose: Verify that Management Endpoint resets are properly isolated.

References:

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: June 15, 2021

Discussion: An SMBus Reset shall be treated by each Command Slot in the 2 Wire Management Endpoint as if an implicit Abort Control Primitive was received with the exception that the Management Endpoint does not transmit the Abort Control Primitive Response Messages

This test is only applicable to products with multiple Management Endpoints within the NVM Subsystem.

Test Setup: See Appendix A.

Test Procedure:
1. Perform MCTP accesses to all management endpoints in the NVM Subsystem.
2. Perform a SMBus Reset on a single SMBus port on the Management Endpoint with the Command Slot in each of the following states:
    a. Idle

b. Receive
c. Process
d. Transmit

Observable Results:

1. Verify that MCTP accesses on any management endpoints not associated with the reset Command Slot completed without error.
2. Verify that in each case, the Command Slot entered the Idle state after the SMBus reset was executed.
3. Verify that the DUT does not transmit the Abort Control Primitive Response.

Possible Problems: None

Test 12.3 – NVMe-MI 2 Wire Reset, bits and fields – (FYI)

Purpose: Verify that Management Endpoint resets are properly handled and that the reset will reset associated bits and fields.

References:

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 23, 2025

Discussion: An SMBus Reset shall be treated by each Command Slot in the 2 Wire Management Endpoint as if an implicit Abort Control Primitive was received with the exception that the Management Endpoint does not transmit the Abort Control Primitive Response Messages. An SMBus Reset shall cause a Management Endpoint Reset of the SMBus/I2C Management Endpoint. A Management Endpoint Reset: resets bits and fields dedicated to the out-of-band mechanism.

Test Setup: See Appendix A.

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.2 or greater, if not skip this test
2. Perform Endpoint Discovery.
3. Perform MCTP accesses to all Management Endpoints in the NVM Subsystem.
4. For each Management Endpoint:
    a. Perform a Controller Health Status Poll Command on the Controller Health Data Structure (CHDS) with the following parameters: Report all set, Include PCI Functions set, Controller Status Changes set, MAXRENT set to 0x01, for the NVMe controller.
    b. Perform a Controller Health Status Poll on the Controller Health Status Changed Flags for the NVMe controller.
    c. Perform a Subsystem Health Status Poll on the NVM Subsystem Health Data structure (NSHDS).
    d. Perform a Configuration Get with Configuration Identifier set to 0x01 (SMBus/I2C Frequency).
    e. Wait for Response message
    f. Perform a SMBus Reset on the controller in the CHDS response.
    g. Perform the same Controller Health Status Poll Commands and NVM Subsystem Health Status Poll Commands as in steps 4.a & 4.b
    h. Perform the same NVM Subsystem Health Status Poll as step 4.c.
    i. Perform a Configuration Get with Configuration Identifier set to 0x01 (SMBus/I2C Frequency)

Observable Results:
1. Verify that MCTP access on any Management Endpoints not associated with the reset Command Slot completed without error.
2. Verify that all Controller and Subsystem Health Status Polls complete successfully.
3. Verify that all fields and bits in the out-of-band mechanism are reset for each Management Endpoint.
4. Verify that all Configuration Get Commands complete successfully and that the SMBus/I2C Frequency is set to 1h (100kHz).

Possible Problems: None

Test 12.4 – NVMe-MI Controller Level Reset – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References:

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: August 1st, 2024

Discussion: A Controller Level Reset : resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. This test is only applicable to a device supporting NVMe-MI 1.2 or greater, if not skip this test
2. Perform Endpoint Discovery.
3. Perform MCTP accesses to all Management Endpoints in the NVM Subsystem.
4. For each Management Endpoint:
    a. Perform a Controller Health Status Poll Command on the Controller Health Data Structure (CHDS) with the following parameters: Report all set, Include PCI Functions set, Controller Status Changes set, MAXRENT set to 0x01, for the NVMe controller.
    b. Perform a Controller Health Status Poll on the Controller Health Status Changed Flags for the NVMe controller.
    c. Perform a Subsystem Health Status Poll on the NVM Subsystem Health Data structure (NSHDS).
    d. Perform a Controller Level Reset on the controller in the CHDS response.
    e. Perform the same Controller Health Status Poll Commands and NVM Subsystem Health Status Poll Commands as in steps 4.a & 4.b.
    f. Perform the same NVM Subsystem Health Status Poll as step 4.c.

Observable Results:
1. Verify that MCTP access on any Management Endpoints is not associated with the reset Command Slot completed without error.
2. Verify that all Controller and Subsystem Health Status Polls complete successfully
3. Verify that all fields and bits in the in-band tunneling mechanisms are reset.
4.

Possible Problems: None

Test 12.5 – Management Endpoint Reset Clears Management Endpoint Buffer – (FYI)

Purpose: Verify that Management Endpoint Rest resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 5.4, 5.5, 5.7, 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: August 1st, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery.
2. For each Management Endpoint:
    a. Read the Port Information Data Structure of the Management Endpoint. Bytes 07:04 indicate the size of the Management Endpoint Buffer in bytes when a Management Endpoint Buffer is supported. A value of 0h in this field indicates that the Management Endpoint does not support a Management Endpoint Buffer, and this test is not applicable.
    b. If Management Endpoint Buffer Size is not 0h, send a Management Endpoint Buffer Write Command to a single Management Endpoint, and fill the buffer with a known data file.
    c. Perform a Management Endpoint Buffer Read to the corresponding buffer.
    d. Perform a Management Endpoint Reset on the corresponding endpoint.
    e. Perform a second Management Endpoint Buffer Read to the corresponding buffer.

Observable Results:
1. Verify that all Management Endpoint Buffer Reads return the data file that was written in step 2b.
2. Verify that all corresponding Management Endpoint Buffers have been cleared to 0h after the reset has completed.

Possible Problems: None

Test 12.6 – Management Endpoint Reset Sets SMBUS/I2C Frequency Field to 1h – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 5.1, 5.2, 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: August 1st, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A

Test Procedure:
1. Perform Endpoint Discovery.
2. For each Management Endpoint:
    a. Perform a Configuration Set (Configuration Identifier 01h) to change the SMBUS/I2C Frequency field to a value other than 1h.
    b. Perform a Configuration Get to confirm the change of value in the SMBUS/I2C Frequency field.
    c. Perform a Management Endpoint Reset.
    d. Perform a Configuration Get.

Observable Results:
1. Verify that all commands complete successfully.

2. Verify that for each Management Endpoint, the Configuration Get from step 2.b returns the value written in step 2a.
3. Verify that the Configuration Get from step 2.d returns 1h after the reset has completed.

Possible Problems: None

Test 12.7 – Management Endpoint Reset sets MCTP Transmission Unit Size field to 40h – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 5.1, 5.2, 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform a Configuration Set (Configuration Identifier 01h) to change the SMBUS/I2C Frequency field to a value other than 1h.
2. Perform a Configuration Read to confirm the change of value in the SMBUS/I2C Frequency field.
3. Perform a Management Endpoint Reset.
4. Perform a Configuration Get.

Observable Results:
1. Verify that all commands completed successfully.
2. Verify that the Configuration Get from step 2 returns the value written in step 1.
3. Verify that the Configuration Get from step 4 returns 40h after the reset has completed.

Possible Problems: None

Test 12.8 – NVM Subsystem Reset Clears Out-of-Band instances of Controller Health Status Changed Flags – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 5.1, 5.2, 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:

1. Perform Endpoint Discovery.
2. Send Configuration Read NVMe-MI Data Structure: with DTYP = 0x00. Record the NVMe-MI NVM Subsystem Capabilities (NNSC) field value. If this bit is set to 0 and the DUT NVMe-MI specification version is later than version 1.2, or if the SRE byte is not set to 01h, this test is not applicable.
3. Perform an NVM Subsystem Reset command.
4. Configure the NVMe Host to issue a Set Feature command to set the Critical Temperature Threshold below the current temperature of the DUT. If Feature Identifier 04h is not supported, skip this step.
5. Issue a Get Log page with LID 02h (SMART/Health Information) and record the value of the Critical Warning byte.
6. For each Management Endpoint, access the instances of the Controller Health Status Changed flags by performing a Controller Health Status Poll command via the out-of-band mechanisms.
7. Perform an NVM Subsystem Reset command.
8. For each Management Endpoint access the instances of the Controller Health Status Changed flags by performing a Controller Health Status Poll command via the out-of-band mechanisms.
9.

Observable Results:

1. Verify all commands complete successfully.
2. For each Management Endpoint verify that the value of the Critical Warning byte from the Get Log Page command is identical to the value returned in the Controller Health Data Structure returned in step 5, and that the CWARN byte is set to 01h.
3. Verify that after the reset which occurs in step 5, each instance of the Controller Health Status Changed flags are cleared to '0'
4.

Possible Problems: None

Test 12.9 – NVM Subsystem Reset Clears All Instances of Composite Controller Status – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 5.6

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:

1. Perform an NVM Subsystem Reset.

2. For each Management Endpoint, perform an NVM Subsystem Health Status Poll command to access the instances of the Composite Controller Status Field, dedicated to both:
    a. The out-of-band mechanism
    b. Each controller in the NVM subsystem

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that for each Management Endpoint, the Subsystem Reset clears the fields with reset values specified for each instance of the Composite Controller Status Field.

Possible Problems: None

Test 12.10 – Controller Level Reset – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Case 1: Controller Level Reset Supports Servicing of Commands

Test Procedure:

1. Place a command from the following command sets on the I/O queue, but do not ring the doorbell:
    a. Management Interface Command Set
    b. NVMe Admin Command Set
    c. Control Primitives
2. Perform a controller level reset (i.e. change CC.EN from ‘1’ to ‘0’, and then back to ‘1’.
3. When the controller level reset is complete, ring the doorbell for the queued commands in step 1.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the commands sent in step 1 continue to be processed during and after the reset.

Possible Problems: None

Case 2: NVMe-MI Command Servicing Unaffected When Another Controller Is Disabled or Being Reset By CLR (FYI)

Test Procedure:

1. This test is only applicable to a device supporting NVMe-MI 2.0 or greater, with multiple controllers in the NVM Subsystem. If this is not the case, then skip this test.
2. Perform Endpoint Discovery.

3. Configure the NVMe Host to disable a single controller in the NVM Subsystem by writing 0b to the Enable (EN) field in the Controller Configuration (CC) register (Offset 14h).
4. Configure the Management Endpoint to issue an NVMe-MI command via the out-of-band mechanism to an enabled controller.
5. Configure the Management Endpoint to issue an NVMe Admin command via the out-of-band mechanism to an enabled controller.
6. If the Management Endpoint supports the use of optional PCIe commands in Figure 148 from the NVMe Management Interface Specification Revision 2.0, configure the Management Endpoint to issue one of the supported commands via the out-of-band mechanism to an enabled controller.
7. Configure the NVMe Host to enable the controller disabled in step 3 by writing 1b to the Enable (EN) field in the Controller Configuration (CC) register (Offset 14h).

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the NVMe-MI command sent in step 4 was unaffected by the other controller being disabled.
3. Verify that the NVMe Admin command sent in step 5 was unaffected by the other controller being disabled.
4. If a PCIe command was issued in step 6, verify that the command was unaffected by the other controller being disabled.

Test 12.11 – Management Endpoint Reset Drops Control Primitive Commands – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:

1. Perform a Control Primitive command (with the Control Primitive Opcode (CPO) set to 00h (Pause).
2. Configure the host to send a Management Endpoint Reset command.

Observable Results:

1. Verify the Reset in step 2 completes successfully.
2. Verify that any Control Primitives sent in step 1 do not return with the Status Code 00h “Success.”

Possible Problems: None

Test 12.12 – Management Endpoint Reset Drops Control Primitive Commands – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 8.3.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform a Control Primitive command (with the Control Primitive Opcode (CPO) set to 00h (Pause).
2. Configure the host to send a Management Endpoint Reset command.

Observable Results:
1. Verify the Reset in step 2 completes successfully.
2. Verify that any Control Primitives sent in step 1 do not return with the Status Code 00h "Success."

Possible Problems: None

Test 12.13 – SMBus Clock-Low Recovery does not reset ARP Values – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 8.1, 8.3.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform an SMBus Reset, via fulfilling clock-low recovery conditions.

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that ARP-assigned addresses have not been reset.

Possible Problems: None

Test 12.14 – 2 Wire Port Remains in Bus Idle Condition during SMBus Reset Assertion – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 8.1, 8.3.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 23, 2025

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform Endpoint Discovery.
2. Perform an SMBus Reset
3. For each Management Endpoint:
    a. Perform an MCTP access to the 2 Wire port.

Observable Results:
1. Verify that the SMBus reset completes successfully.
2. For each Management Endpoint, verify that the port used in this test supports 2 Wire port accesses starting 1 second after the de-assertion of the reset.

Test 12.15 – SMBus Reset Resets the Value of the SMBus/I2C Frequency Field – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 5.1.1, 5.2.1, 8.3.4

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: August 1st, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform endpoint discovery.
2. For each Management Endpoint:
    a. Perform a Configuration Set command to change the SMBus/I2C frequency field to a value other than reset default (01h).
    b. Perform a Configuration Get Command.
    c. Perform a SMBus Reset.
    d. Perform a Configuration Get command.

Observable Results:
1. Verify that all commands complete successfully.
2. For each Management Endpoint verify that the SMBus/I2C frequency field has changed to a value other than reset default from step 2.
3. For each Management Endpoint, verify that the Configuration Get command returns the reset default value (01h) for SMBus/I2C frequency field post-reset.

Test 12.16 – .PCIe VDM Management Endpoint supports PCIe MCTP Access Post-Reset – (FYI)

Purpose: Verify that Controller Level resets are properly handled and that the reset will reset associated bits and fields.

References: NVMe-MI Specification v1.2d sections 8.3

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 18, 2024

Discussion: A Controller Level Reset: resets bits and fields dedicated to the in-band tunneling mechanisms.

Test Setup: See Appendix A.

Test Procedure:
1. Perform a PCIe reset.
2. Perform an MCTP access to the PCIe VDM Management Endpoint.

Observable Results:
1. Verify all commands complete successfully.
2. Verify that the PCIe VDM Management Endpoint supports PCIe MCTP access 1 second after the de-assertion of the reset.

Group 13: Shutdown Processing

Case 1: Get Log Page SMART / Health Information when shutdown is Occurring (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later,, continue with step 3, otherwise this test is not applicable.
3. Perform a get log on SMART / Health Information (02h) and store the Unsafe Shutdowns value
4. Perform a shutdown, where the main power is lost.
5. If CSTS.SHST is set to 10b, perform a get log on SMART / Health Information (02h) and store the Unsafe Shutdowns value.
6. Perform an Admin Command with the Ignore Shutdown bit set to 1, then perform a shutdown where the main power is lost (CSTS.SHST is set to 01b or 10b).
7. If CSTS.SHST is set to 01b or 10b, perform a get log on SMART / Health Information (02h) and store the Unsafe Shutdowns value.

Observable Results:

1. Verify that the Unsafe Shutdown field increments after procedure steps 3 & 5 & 7.

Case 2: CSTS.SHST when shutdown is Occurring (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later, continue with step 3, otherwise this test is not applicable.
3. Perform a shutdown on the NVMe controller.
4. Perform a read on the CSTS.SHST field and record the value.
5. Perform an Admin Command with the Ignore Shutdown (ISH) bit set to 1.
6. Perform an Admin Command with the Ignore Shutdown (ISH) bit set to 0.
7. Perform a read on the CSTS.SHST field and record the value

Observable Results:

1. Verify that CSTS.SHST field does not change due to Admin Commands with the Ignore Shutdown set to 0 or 1.

Case 3: Shutdown, ISH Setting, No Change in CSTS.SHST (FYI)

Test Procedure:

1. Perform Endpoint Discovery.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later,, continue with step 3, otherwise this test is not applicable.
3. Perform a shutdown on the NVMe controller.
4. Perform a read on the CSTS.SHST field and record the value.
5. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 1.
6. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 0.
7. Perform a read on the CSTS.SHST field and record the value

Observable Results:

1. Verify that CSTS.SHST field does not change due to Admin Commands with the Ignore Shutdown set to 0 or 1.

Case 4: Shutdown, Admin Command Aborted, CSTS.SHST=00b (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later,, continue with step 3, otherwise this test is not applicable.
3. Read the CSTS.SHST field and verify it is set to 00b, if not end this test case
4. Perform a shutdown on the NVMe controller.
5. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 1.
6. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 0.

Observable Results:

1. Verify that both Admin Commands are aborted with the Status field in the Completion Queue Entry Dword 3 is set to a value of "Command Aborted due to Power Loss Notification".

Possible Problems: The device may not abort the NVMe Admin Command.

Case 5: Shutdown, Admin Command Aborted, CSTS.SHST=01b, ISH=0 (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported. If the DUT supports NVMe-MI version 1.2 or later,, continue with step 3, otherwise this test is not applicable.
3. Read the CSTS.SHST field and verify it is set to 01b, if not end this test case
4. Perform a shutdown on the NVMe controller.
5. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 0.

Observable Results:

1. Verify that the Admin Command is aborted with the Status field in the Completion Queue Entry Dword 3 is set to a value of "Command Aborted due to Power Loss Notification".

Possible Problems: The device may not abort the NVMe Admin Command.

Case 6: Shutdown, Admin Command Aborted, CSTS.SHST=01b, ISH=1 (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported If the DUT supports NVMe-MI version 1.2 or later,, continue with step 3, otherwise this test is not applicable.
3. Read the CSTS.SHST field and verify it is set to 01b, if not end this test case
4. Perform a shutdown on the NVMe controller.
5. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 1.

Observable Results:

1. Verify that the Admin Command is processed normally.

Case 7: Shutdown, Admin Command Aborted, CSTS.SHST=10b, ISH=0 (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Read the NVMe Subsystem Information Data Structure to get the MJR&MNR version supported If the DUT supports NVMe-MI version 1.2 or later,, continue with step 3, otherwise this test is not applicable.
3. Read the CSTS.SHST field and verify it is set to 10b, if not end this test case
4. Perform a shutdown on the NVMe controller.
5. Perform an Admin Command (SMART / Health Information, LID 02h) with the Ignore Shutdown bit set to 0.

Observable Results:

1. Verify that the Admin Command is aborted with the Status field in Completion Queue Entry Dword 3 is set to a value of Admin Command Media Not Ready.

Possible Problems: The Admin Command might take longer than normal to be performed.

Group 14: Asynchronous Events

Case 1: Response Message ROR=1 (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Configure the NVMe Host to issue a Get Log Page command as an out-of-band Request Message to the DUT.

Observable Results:

1. Verify that the Request or Response (ROR) bit of the returned NVMe-MI Message Parameters (NMP) field from step 2 is set to 1.

Case 2: NMINT Expected Return Value (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Send a Request Message to the DUT where the expected Response Message will have the value 0h (Control Primitive) in the NVMe-MI Message Type (NMINT) field.
3. Send a Request Message to the DUT where the expected Response Message will have the value 1h (NVMe MI Command) in the NVMe-MI Message Type (NMINT) field.
4. Send a Request Message to the DUT where the expected Response Message will have the value 2h (NVMe Admin Command) in the NVMe-MI Message Type (NMINT) field.
5. Send a Request Message to the DUT where the expected Response Message will have the value 4h (PCIe Command) in the NVMe-MI Message Type (NMINT) field.
6. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
7. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable a supported AE from the previous step with an AEM Delay of 5 seconds.
8. Configure the NVMe Host to trigger the enabled AE.
9. Delay for at least 5 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that the NVMe-MI Message Type field in all of the Response Messages are the expected values.
2. Verify that the NVMe-MI Message Type field in the AEM transmitted after step 10 contains a value of 5h (Asynchronous Event)

Case 3: Out of Band Response Message Command Slot Identifier (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Send an out-of-band Request Message to the DUT where the Command Slot Identifier Field of the Request Message is 0b.
3. Send an out-of-band Request Message to the DUT where the Command Slot Identifier Field of the Request Message is 1b

Observable Results:

1. Verify that the Response Message returned in step 2 has a Command Slot Identifier Field value of 0b.
2. Verify that the Response Message returned in step 3 has a Command Slot Identifier Field value of 1b.

Case 4: Command Slot Identifier Field Not Applicable to AEMs (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to all supported AEs which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 20 seconds.

4. Configure the NVMe Host to trigger all enabled AEs.
5. Delay for at least 20 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that the AEM received after step 5 contains a Command Slot Identifier field value of 0h for all AEs triggered in step 4

Case 5: Request Message Integrity Check Bit =1b (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Configure the NVMe Host to issue a Get Log Page command as an out-of-band Request Message to the DUT with the Integrity Check bit set to 1b.
3. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
4. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable one or more supported AEs which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
5. Configure the NVMe Host to trigger an enabled AE.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that the returned Response Message to the Message sent in step 2 has the Integrity Check bit in the MCTP Data Field set to 1b.
2. Verify that the returned Response Message to the Message sent in step 2 contains a CRC computed over the contents of the NVMe-MI Message.
3. Verify that the AEM received in the AEM Transmission Interval has the Integrity Check bit in the MCTP Data Field set to 1b.
4. Verify that the AEM also contains a CRC computed over the contents of the NVMe-MI Message.

Case 6: Request Message Integrity Check Bit=0b (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Configure the NVMe host to issue a Request Message as a Configuration Set command to the DUT with an Integrity Check bit value of 0b

Observable Results:

1. Verify that the returned Response Message to the Message sent in step 2 has an Integrity Check bit value of 0b.
2. Verify that the returned Response Message to the Message sent in step 2 does not have a Message Integrity Check field.

Case 7: Multi Packet Response Message MCTP Transmission Unit Size (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Configure the NVMe Host to issue a Get Log Page command as a Request Message to the DUT

Observable Results:

1. Verify that all packets except for the last packet sent in step 2 are equal to the negotiated MCTP Transmission Unit Size

Case 8: Invalid MIC Value Request Message (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Send an out-of-band Request Message to the DUT with an invalid MIC value.

3. Wait for 60 seconds.

Observable Results:

1. Verify that the Request Message is discarded and that no Response Message is received before the completion of step 3

Case 9: Management Endpoint AEM Support (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Perform Endpoint Discovery.
3. Configure the NVMe host to issue Configuration Get commands with Configuration Identifier 04h to each management endpoint with the Asynchronous Event Messages Supported bit set to 01h

Observable Results:

1. Verify that each AE Supported List Data Structure returned from the Configuration Get Commands sent in step 2 specifies at least one supported AE.

Case 10: Management Endpoint Buffer Command Message Error Handling (FYI)

Test Procedure:

1. Perform MCTP initialization.
2. Configure the NVMe Host to retrieve the Management Endpoint Buffer Command List Data Structure.
3. Configure the NVMe Host to issue a command found in the previous step with the Management Endpoint Buffer (MEB) bit set to 1b in a Command Message that contains Request Data.

Observable Results:

1. Verify that the Management Endpoint responds with an Invalid Parameter Error Response with the PEL field indicating the Request Data Field of the Command Message in response to the executed command in step 3.

Case 11: Management Endpoint Buffer Command Message 0h Offset (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to retrieve the Management Endpoint Buffer Command List Data Structure.
3. Configure the NVMe Host to issue a command found in the previous step with the Management Endpoint Buffer (MEB) bit set to 1b in a Command Message that does not contain Request Data.

Observable Results:

1. Verify that the Request Data for the Command Message issued in step 3 has Request Data starting from a 0 offset from the start of the Management Endpoint Buffer.

Case 12: Asynchronous Event Disarmed State (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Perform Endpoint Discovery.
3. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
4. Configure the NVMe Host to issue Configuration Set commands with Configuration Identifier 04h to disable all Asynchronous Events on all Management Endpoints for the DUT.
5. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that no AEs are enabled in the AE Enable List received in step 5.

Case 13: Asynchronous Event Message Delay Interval (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and an AEM Delay of 5 seconds to enable one or more supported AEs from the previous step.
4. Configure the NVMe Host to trigger an enabled AE.
5. Delay 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that no AEM is received during the 5 second delay in step 5.
2. Verify that an AEM is received after step 5.

Case 14: Asynchronous Event Delay Interval - AE Disarm (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and an AEM Delay value of 5 to enable one or more supported AEs from the previous step.
4. Configure the NVMe Host to trigger an enabled AE.
5. Configure the NVMe host to issue a Configuration Set command with Configuration Identifier 04h to disable all supported AEs.
6. Delay for 5 seconds to simulate entering the AEM Transmission Interval.

Observable Results:

1. Verify that no AEM is received after the delay in step 6.

Case 15: AEM Transmission Interval - AE Sync (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and an Asynchronous Event Message Delay value of 5 seconds to enable a supported AE from the previous step.
4. Configure the NVMe Host to trigger the AE specified in the previous step.
5. Delay for 5 seconds to enter the AEM Transmission Interval.
6. Immediately configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to disable the AE enabled in step 3 and trigger an AE Sync.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that no AEM containing the AE triggered in step 4 is received, i.e., the AEM is discarded.

Case 16: AEM Transmission Interval - AE Sync (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and an Asynchronous Event Message Delay value of 5h to enable a supported AE from the previous step.
4. Configure the NVMe Host to trigger an enabled AE from the previous step.
5. Delay for 5 seconds to enter the AEM Transmission Interval.

6. Immediately configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04 and the Number of AE Enable Data Structures cleared to 0h to trigger an AEM Ack.

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that no AEM containing the AE Triggered in step 5 is received, i.e., the AEM is discarded

Case 17: AEM Transmission Interval - AEM Transmission Failure (FYI)

Test Procedure:
1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and an Asynchronous Event Message Delay of 5 seconds and an AEM Retry Delay value of 1h to enable a supported AE from the previous step.
4. Configure the NVMe Host to trigger an enabled AE from the previous step.
5. Configure the NVMe Host to not receive AEMs during the AEM Transmission Interval.
6. Delay for 6 seconds to enter the AEM Transmission Interval.
7. Configure the NVMe Host to issue a NVM Subsystem Health Poll command.

Observable Results:
1. Verify that all commands completed successfully.
2. Verify that the ATF bit is set to 1 in the NVM Subsystem Health Data Structure received in step 7.
3. Verify that no AEM is received after the AEM Transmission Interval has completed.

Case 18: AEM Transmission Interval - Management Endpoint Reset (FYI)

Test Procedure:
1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and an Asynchronous Event Message Delay of 5 seconds and an AEM Retry Delay value of 1h to enable a supported AE from the previous step.
4. Configure the NVMe Host to trigger an enabled AE from the previous step.
5. Delay for 5 seconds to enter the AEM Transmission Interval.
6. Immediately reset the Management Endpoint.

Observable Results:
1. Verify that all commands completed successfully.
2. Verify that an AEM containing the AE triggered in step 4 is not received before or after the reset in step 6

Case 19: AEM Retry-AE Sync (FYI)

Test Procedure:
1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable one or more supported AEs which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds and an AEM Retry Delay value of 1h.
4. Configure the NVMe Host to trigger an enabled AE from the previous step.
5. Configure the NVMe Host to not receive AEMs during the AEM Transmission Interval.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

7. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h and the Number of AE Enable Data Structures field not cleared to 0h to trigger an AE Sync.

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that the Management Endpoint no longer retries the AEM Transmission after the AE sync in step 7, i.e. no AEM is received for the AE triggered in step 4.

Case 20: AEM Retry-AEM Ack (FYI)

Test Procedure:
1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable one or more supported AEs which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds and an AEM Retry Delay of 10h.
4. Configure the NVMe Host to trigger an enabled AE from the previous step.
5. Configure the NVMe Host to not receive AEMs.
6. Delay for 5 seconds to enter the AEM Transmission Interval.
7. Delay for 1 second to initiate a retry.
8. Configure the NVMe Host to receive AEMs.
9.

Observable Results:
1. Verify that the Management Endpoint retries the AEM Transmission after the failure between steps 6 and 7.
2. Verify that the Management Endpoint no longer retries the AEM Transmission after the AEM ACK is returned in response to the AEM received after step 8 (i.e. no other AEMs are received).
3. Verify that the AEM contains an AEM Retry Count greater than 0.

Case 21: AEM Retry-Management Endpoint Reset (FYI)

Test Procedure:
1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable one or more supported AEs which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay for 5 seconds with an AEM Retry Delay of 1h.
4. Configure the NVMe Host to trigger an enabled AE from the previous step.
5. Configure the NVMe Host to not receive AEMs.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.
7. Reset the Management Endpoint.

Observable Results:
1. Verify that the Management Endpoint no longer retries the AEM Transmission after the Management Endpoint Reset that occurred in step 7

Case 22: AEM Retry AE Occurrence Data Structure Isolation (FYI)

Test Procedure:
1. Perform MCTP Initialization.
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) with an AEM Retry value of 5 seconds to enable a supported event found in the previous step.

4. Configure the NVMe Host to trigger an enabled Asynchronous Event from the previous step.
5. Configure the NVMe Host to not receive AEMs.
6. Delay 5 seconds to enter the AEM Transmission Interval.
7. Configure the NVMe Host to trigger another enabled Asynchronous Event.
8. Configure the NVMe Host to receive AEMs

Observable Results:

1. Verify that an AEM is received after step 8 and contains an AEM Retry Count value greater than 0, the AE triggered in step 4, and does not contain the AE triggered in step 7

Case 23: Msg Tag, Owner Bit, Destination Endpoint ID (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable a supported event found in the previous step with an AEM Retry value of 5 seconds to enable a supported event found in the previous step.
4. Configure the NVMe Host to trigger an enabled Asynchronous Event specified in the previous step.
5. Delay for at least 5 seconds to enter the AEM Transmission interval

Observable Results:

1. Verify that the Msg tag for the AEM received after step 4 is selected by the Management Endpoint.
2. Verify that the Tag Owner Bit for the AEM is set to '1'.
3. Verify that the Destination Endpoint ID of the AEM is equal to the value of the Source Endpoint ID of the NVMe Host (which is the entity that most recently caused an AE Arm in step 3).

Case 24: Occurrence Namespace ID Field (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable a supported event within the Namespace Scope found in the previous step and set an AER Delay value of 5 seconds. If no Namespace Scope events are supported, this test is not applicable.
4. Configure the NVMe Host to trigger an enabled AE specified in the previous step.
5. Delay for at least 5 seconds to enter the AEM Transmission Interval.
6. Record the AE Occurrence Scope ID Info (AEOCIDI) associated with the AE Occurrence generated from the previous step.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the AE Occurrence Namespace ID field (AEONID) matches the NSID of the namespace associated with the AE triggered in step 4

Possible Problems: As of NVMe-MI Specification 1.2d, this test will always be skipped as there are no Asynchronous Events that fall under the Namespace Scope.

Case 25: Occurrence Controller ID Field (FYI)

Test Procedure:

1. Perform MCTP Initialization

2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable a supported event within the Controller Scope found in the previous step and set an AER Delay value of 5 seconds. If no Controller Scope events are supported, this test is not applicable.
4. Configure the NVMe Host to trigger an enabled AE specified in the previous step.
5. Delay for at least 5 seconds to enter the AEM Transmission Interval.
6. Record the AE Occurrence Scope ID Info (AEOCIDI) associated with the AE Occurrence generated from the previous step.

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that the AE Occurrence Controller ID field (AEOCID) matches the Controller ID of the controller associated with the AE triggered in step 4

Case 26: AE Occurrence Endurance Group ID (FYI)

Test Procedure:
1. Perform MCTP Initialization
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable a supported event found in the previous step and set an AER Delay value of 5h. If no Endurance Group Scope events are supported, this test is not applicable.
4. Configure the NVMe Host to trigger an enabled AE specified in the previous step.
5. Delay for at least 5 seconds to enter the AEM Transmission Interval.
6. Record the AE Occurrence Scope ID Info (AEOCIDI) associated with the AE Occurrence generated from the previous step. If the AE Occurrence Scope (AESS) field indicates a value other than 5h, then this test is not applicable.

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that the AE Occurrence Group ID (AEOEGID) field contains the Endurance Group ID of the Endurance Group associated with the AE.

Possible Problems: As of NVMe-MI Specification 1.2d, this test will always be skipped as there are no Asynchronous Events that fall under the Endurance Group Scope.

Case 27: Controller Ready AE (00h) (FYI)

Test Procedure:
1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Controller Ready AE is not included in the AE Supported List data structure then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Controller Ready AE (00h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5h.
4. Configure the NVMe Host to trigger the enabled Controller Ready AE from step 3.
5. Configure the NVMe Host to read from Controller Register Offset 1Ch (Controller Status) and record the RDY bit.
6. Delay for at least 5 seconds to start the AEM Transmission Interval

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that the Ready Value bit in the AE Occurrence Specific Info Data Structure from step 4 is the same as the CSTS.RDY bit

Case 28: Controller Fatal Status AE (01h) (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If Controller Fatal Status AE is not included in the AE Supported List data structure then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Setcommand with Configuration Identifier 04h to enable the Controller Fatal Status AE (01h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Controller Fatal Status AE from step 3.
5. Configure the NVMe Host to read from Controller Register Offset 1Ch (Controller Status) and record the CFS bit.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the Controller Fatal Status Value in the AE Occurrence Specific Info Data Structure from step 4 is the same as the CSTS.CFS bit.

Case 29: Shutdown Status AE (02h) (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Shutdown Status AE is not included in the AE Supported List data structure then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Shutdown Status AE (02h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Shutdown Status AE from step 3.
5. Configure the NVMe Host to read from Controller Register Offset 1Ch (Controller Status) and record the CSTS.ST and CSTS.SHST bit.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that the Shutdown Type Value and the Shutdown Status Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the CSTS.ST and CSTS.SHST value respectively

Case 30: Controller Enable AE (03h) (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Controller Enable AE is not included in the AE Supported List data structure then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Controller Enable AE (03h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Controller Enable AE from step 3.
5. Configure the NVMe Host to read from Controller Register Offset 00h (Controller Capabilities) and record the CC.CRIME bit and the CC.EN bit.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that the Controller Ready Independent of Media Enable value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the CC.CRIME value.

2. Verify that the Controller Enable Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as CC.EN value.

Case 31: Namespace Attribute Changed AE (04h) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Namespace Attribute Changed AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Namespace Attribute Changed AE (04h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Namespace Attribute Changed AE from step 3.
5. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that there is no AE Occurrence Specific Info for the Namespace Attribute Changed AE triggered in step 4.

Case 32: Firmware Activated AE (05h) (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Controller Enable AE is not included in the AE Supported List data structure then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Controller Enable AE (03h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Controller Enable AE from step 3.
5. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Firmware Activated AE (05h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
6. Configure the NVMe Host to trigger the enabled Firmware Activated AE from step 3.
7. Delay for at least 5 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that there is no AE Occurrence Specific Info for the Firmware Activated AE triggered in step 4.

Case 33: Composite Temperature AE (06h) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Composite Temperature AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Composite Temperature AE (06h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Composite Temperature AE from step 3.
5. Configure the NVMe Host to issue a Subsystem Health Status Poll on the NVM Subsystem Health Data Structure.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that the Composite Temperature Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the Composite Temperature field in the NVM Subsystem Health data structure.

Case 34: Percentage Drive Life Used AE (07h) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Percentage Drive Life Used AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Percentage Drive Life Used AE (07h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Percentage Drive Life Used AE from step 3.
5. Configure the NVMe Host to issue a Subsystem Health Status Poll on the NVM Subsystem Health Data Structure.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that the Percentage Drive Life Used Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the Percentage Used field in the NVM Subsystem Health data structure.

Case 35: Available Spare AE (08h) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Available Spare AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Available Spare AE (08h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Available Spare AE from step 3.
5. Configure the NVMe Host to issue a Subsystem Health Status Poll on the NVM Subsystem Health Data Structure.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that the Available Spare Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the Available Spare field of any Controller in the NVM Subsystem.

Case 36: SMART Warnings AE (09h) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the SMART Warnings AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the SMART Warnings AE (09h) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled SMART Warnings AE from step 3.
5. Configure the NVMe Host to issue a Subsystem Health Status Poll on the NVM Subsystem Health Data Structure.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that the SMART Warnings Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the value of the SMART Warnings field in the NVM Subsystem Health Data Structure.

Case 37: Telemetry Controller-Initiated Data Available AE (0Ah) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Telemetry Controller-Initiated Data Available AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Telemetry Controller-Initiated Data Available AE (0Ah) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Telemetry Controller-Initiated Data Available AE from step 3.
5. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that there is no AE Occurrence Specific Info for the Telemetry Controller-Initiated Data Available AE triggered in step 4.

Case 38: PCIe Link Active AE (0Bh) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the PCIe Link Active AE is not included in the AE Supported List data structure then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the PCIe Link Active AE (0Bh) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled PCIe Link Active AE from step 3.
5. Configure the NVMe Host to issue a Subsystem Health Status Poll on the NVM Subsystem Health Data Structure.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.
7. Record the AE Occurrence Scope ID Info (AEOCIDI) associated with the AE Occurrence generated from the PCIe Link Active AE to retrieve the Port Scope ID Info field.

Observable Results:

1. Verify that the PCIe Link Active Value from the AE Occurrence Specific Info Data Structure from step 4 is the same as the value of the PCIe Port 0 PCIe Link Active bit in the NVM Subsystem Health Data Structure or the PCIe Port 1 PCIe Link Active bit in the NVM Subsystem Health Data Structure depending on if the value of the Port Scope ID Info Field from step 6 indicates the AE is associated with PCIe Port 0 or PCIe Port 1.

Case 39: Sanitize Failure Mode AE (0Ch) (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure. If the Sanitize Failure Mode AE is not included in the AE Supported List data structure, then this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable the Sanitize Failure Mode AE (0Ch) which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled Sanitize Failure Mode AE from step 3.
5. Configure the NVMe Host to issue a Subsystem Health Status Poll on the NVM Subsystem Health Data Structure.
6. Delay for at least 5 seconds to enter the AEM Transmission Interval.

Observable Results:

1. Verify that the Sanitizer Failure Mode Value from the AE Occurrence Specific Info Data Structure from the AEM after step 6 is the same as the value of the Sanitize Failure Mode bit in the NVM Subsystem Health Data Structure.

Case 40: AE Supported List Data Structure (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.

Observable Results:

1. Verify that the AEELVER value returned in the NVMe Management Response from the Configuration Get command is cleared to 0h.
2. Verify that the AE Supported List Data Structure is valid by checking the following values:
    a. NUMAES >= 1h,
    b. AESLVER = 0h,
    c. AESTL equal to the total length of the data structure,
    d. AESLHL = 5h,
    e. For each AE Supported in the AE Supported List Body, AESL = 3h

Case 41: AE Sync Processing (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.
3. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event), Number of AE Enable Data Structures (NUMAEE) set to 1 and valid AE Enable data structure using a supported event found in the previous step

Observable Results:

1. Verify that the AE Occurrence List Overflow bit returned in step 3 is cleared to 0.

Case 42: AEM Ack During AE Disarmed State (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable a supported event found in the previous step.
4. Configure the NVMe Host to trigger an enabled AE specified in the previous step.
5. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to disable all AEs that were reported as enabled in the previous step.
6. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) and Number of AE Enable Data Structures (NUMAEE) cleared to 0h.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the AE Occurrence List does not contain the AE Occurrence triggered in step 4.
3. Verify that the AE Occurrence List Overflow bit is cleared to 0.

Case 43: AE Enable List Body > 4 KiB (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h (Asynchronous Events) and an AE Enable List Body greater than 4 KiB

Observable Results:

1. Verify that the Management Endpoint responds to the Configuration Set with a Response Message Status of Invalid Command Input Data Size.

Case 44: AE Sync AEM Not Sent (FYI)

Test Procedure:

1. Perform MCTP Initialization
2. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.
3. If there are no events enabled, configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) and an AE Enable List Data Structure configured to enable one of the supported events discovered in the previous step with an AEM Delay of 5 seconds.
4. Configure the NVMe Host to trigger the enabled AE from the previous step.
5. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h (Asynchronous Event) and Number of AE Enable Data Structures (NUMAEE) not cleared to 0h to process an AE Sync.
6. Delay for at least 5 seconds to simulate entering the AEM Transmission Interval

Observable Results:

1. Verify that all commands complete successfully.
2. Verify that the AE Occurrence List from step 5 does not contain any events.
3. Verify that no AEM is received after step 6.

Case 45: AE Transmission Interval Discard AE Armed (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to enable one or more supported AEs which will trigger an AE Arm. Also configure NVMe Management Dword 0 to set an AEM Delay of 20h.
4. Configure the NVMe Host to trigger an enabled AE to start the AEM Transmission Interval.
5. Configure the NVMe Host to issue a Configuration Set command with NUMAEE cleared to 0h to process an AEM Ack.
6. Delay for at least 20 seconds to enter the AEM Transmission Interval

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that the AE triggered in step 5 was not present in the AEM Ack from step 6.

Case 46: AE Occurrence List AE 4KiB Overflow (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. If the Management Endpoint Buffer (MEB) bit is set to ‘1’, this test is not applicable.
3. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List data structure.
4. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to Enable all supported Asynchronous Events.
5. Configure the NVMe Host to generate enough Asynchronous Events such that the length of the AE Occurrence List Body exceeds 4 KiB.

6. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h to retrieve the AE Occurrence List Body within the Response Data field of the Response Message.
7. Configure the NVMe Host to trigger an AE Sync to resynchronize the state of AEs between the Management Controller and the Management Endpoint

Observable Results:
1. Verify that all commands completed successfully.
2. Verify that the AE Occurrence List Overflow field within AEOLLI (bytes 4:2) of the AE Occurrence List Data Structure is set to 1.

Case 47: AE Occurrence List Data Structure Response Message (FYI)

Test Procedure:
1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h to retrieve the AE Supported List Data Structure.
3. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h (Asynchronous Event) and an AE Enable List Data Structure configured to enable or disable a supported event.

Observable Results:
1. Verify that all commands completed successfully.
2. Upon completion of the Configuration Set command, verify that a Response Message is transmitted.
3. Verify an AE Occurrence List Data Structure is included within the Response Data field of the Response Message, and that it begins at byte 4 of the NVMe-MI Message field.

Case 48: AE Enable List Supported Event Not Included (FYI)

Test Procedure:
1. Perform MCTP Initialization.
2. Configure the NVMe Host to issue a Configuration Get command with Configuration Identifier 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.
3. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) and an AE Enable List Data Structure configured to enable or disable one of the supported events discovered in the previous step.
4. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.

Observable Results:
1. Verify that all commands complete successfully.
2. Verify that the state of the configuration remains the same in steps 2 and 4, except for the event enabled or disabled in step 3.

Case 49: AE Enable List Unsupported Event Included (FYI)

Test Procedure:
1. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.
2. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) to enable an unsupported AE not found in the previous step.
3. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.

Observable Results:
1. Verify all commands completed successfully.
2. Verify that the configuration states recorded in step 2 and step 4 are the same.

Case 50: Configuration Set No Asynchronous Events Enabled (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.
3. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) to disable any enabled events discovered in the previous step.
4. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) and Number of AE Enable Data Structures (NUMAEE) cleared to 0

Observable Results:

1. Verify that all Configuration Set commands completed successfully.
2. Verify that a Success Response Message is received after the second Configuration Set command.

Case 51: Configuration Set AE Armed State (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe host to issue a Configuration Get command with a Configuration Identifier of 04h (Asynchronous Event) to retrieve the AE Supported List Data Structure. Record the state of the configuration.
3. If there are no events enabled, configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) and an AE Enable List Data Structure configured to enable one of the supported events discovered in the previous step.
4. Configure the NVMe Host to issue a Configuration Set command with Configuration Identifier 04h (Asynchronous Event) and Number of AE Enable Data Structures (NUMAEE) cleared to 0h to process an AEM Ack.
5. Configure the NVMe Host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event)

Observable Results:

1. Verify that all Configuration Set commands complete successfully.

Case 52: AE Occurrence Data Structure During AE Disarmed State and AE enabled (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable one of the supported AEs recorded in the previous step.
4. Configure the NVMe Host to trigger the enabled AE recorded in step 2 to start the AEM Transmission Interval.
5. Configure the NVMe Host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event)

Observable Results:

2. Verify all commands completed successfully.
3. Verify the response from step 5 only contains an AE Occurrence Data Structure for the AE specified in step 4.

Case 53: AE Occurrence Data Structure During AE Disarmed State (FYI)

Test Procedure:

1. Perform MCTP Initialization.

2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs. Record the state configuration.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable two different AEs that were reported as supported in the previous step.
4. Configure the NVMe Host to trigger an enabled AE recorded in step 2 to start the AEM Transmission Interval.
5. Configure the NVMe Host to trigger a different enabled AE recorded in step 2 to start the AEM Transmission Interval.

Observable Results:

1. Verify the unique ID returned in the second AEM is the unique ID associated with the second triggered AE from step 5, and does not contain any other AEM IDs.

Case 54: AE Occurrence List Data Structure after AE Sync (FYI)

Test Procedure:

1. Perform MCTP Initialization.
2. Configure the NVMe Host to perform a Configuration Get command to Configuration Identifier 04h (Asynchronous Event) to receive the list of supported AEs.
3. Configure the NVMe Host to perform a Configuration Set command to Configuration Identifier 04h (Asynchronous Event) to enable all supported AEs with an AEM Delay value of 5h.
4. Configure the NVMe host to issue a Configuration Set command with a Configuration Identifier of 04h (Asynchronous Event) and Number of AE Enable Data Structures (NUMAEE) set to the number of events enabled in step 3

Observable Results:

1. Verify all commands completed successfully.
2. Verify the AE Occurrence List Data Structure contains an AE Occurrence Data Structure for each AE that was enabled by the Configuration Set command in step 3.
3. Verify that each AE Occurrence Data Structure returned in step 4 indicates the state of each AE at the time the Configuration Set command in step 4 was processed.

Group 15: I3C Modifications

Test 15.1 – I3C Modifications – (FYI)

Purpose: Check operation and fields in I3C mode per TP6037.

References:
NVMe-MI Specification 2.0

Resource Requirements:
Tools capable of stimulus and decoding of traffic on the MCTP and NVMe-MI interface.

Last Modification: January 23 ,2025

Discussion: The NVM Express Management Interface has added support for I3C in NVMe Management Interface 2.0. The following tests verify the correct behavior is implemented in accordance with the specification.

Test Setup: See Appendix A.

Case 1: I3C Provisioned ID Field Least Significant Bits Match UDID Devic ID (FYI)

Test Procedure:
1. Perform MCTP initialization.
2. Verify that SMBus ARP is supported by checking that the ARP Capable field in the NVM Subsystem Element Descriptor is set to 1. If the ARP Capable field is cleared to 0 then this test is not applicable.
3. Check the least 30 significant bits of the I3C Provisioned ID field.
4. Check the UDID Device ID.

Observable Results:
1. Verify that the least 30 significant bits of the I3C Provisioned ID field and the UDID Device ID are equal.

Possible Problems: None

Case 2: MCTP Transmission Unit Size Command when 2-Wire port is in I3C Mode (FYI)

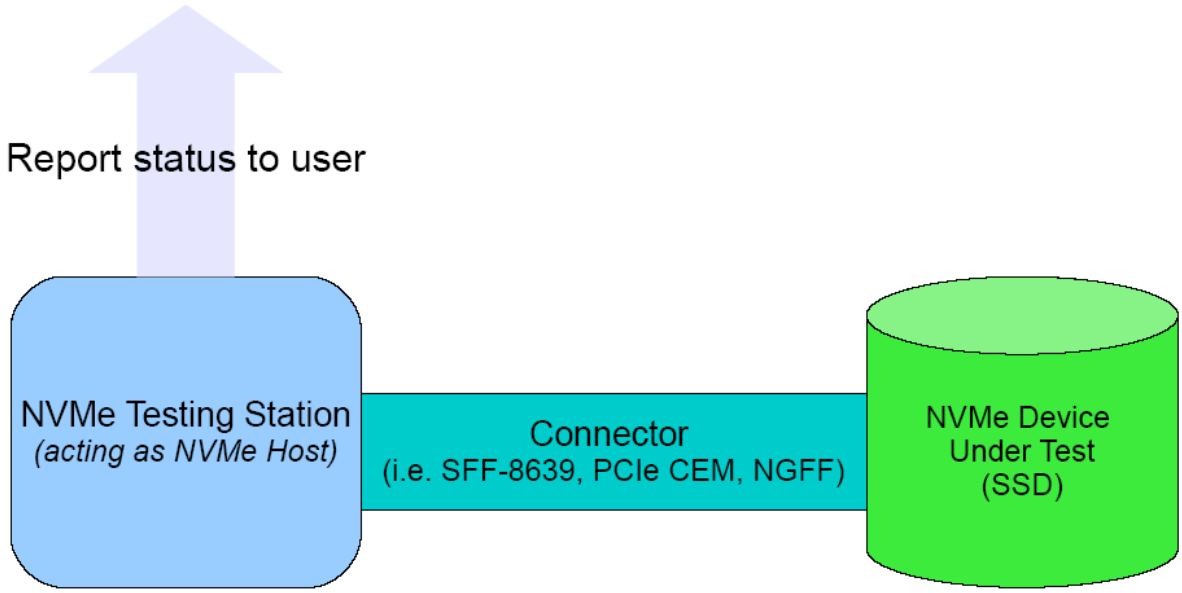
Test Procedure:
1. Perform MCTP initialization.
2. Configure the 2-Wire port to be in I3C mode.
3. Configure the NVMe Host to issue an MCTP Transmission Unit Size Command to the DUT.

Observable Results:
1. Verify that the command from step 3 completes successfully.

Possible Problems: None

Appendix A: Default Test Setup

Except where otherwise specified, all tests will require the DUT to have one of the following default physical configuration at the beginning of each test case:



Appendix B: Notes on Test Procedures

There are scenarios where in test procedures it is desirable to leave certain aspects of the testing procedure as general as possible. In these cases, the steps in the described test procedure may use placeholder values, or may intentionally use non–specific terminology, and the final determination of interpretation or choice of values is left to the discretion of the test technician. The following is an attempt to capture and describe all such instances used throughout the procedures.

Ports on Testing Station and Device Under Test

In general, any PCIe Port on the Testing Station or Device Under Test may be used as an interface with a test station or interoperability partner. There is *assumed* to be no difference in behavior, with respect to the protocols involved in this test suite, between any two PCIe ports on the Testing Station or Device Under Test. Hence, actual ports used may be chosen for convenience. However, it is recommended that the PCIe port used in the test configuration is recorded by the test technician.

Use of "various"

To maintain generality, some steps will specify that "various other values" (or the like) should be used in place of a given parameter. Ideally, all possible values would be tested in this case. However, limits on available time may constrain the ability of the test technician to attempt this. Given this, a subset of the set of applicable values must generally be used.
When deciding how many values should be used, it should be noted that the more values that are tested, the greater the confidence of the results obtained (although there is a diminishing return on this).
When deciding which specific values to use, it is generally recommended to choose them at pseudo–randomly yet deterministically. However, if there exist subsets of the applicable values with special significance, values from each subset should be attempted.

Appendix C: TEST TOOLS

The Tests described in this document can be performed using available Teledyne-LeCroy Test Tools. These tests are supported in v8.77 or higher of the Teledyne-LeCroy PC Edition software available at: http://teledynelecroy.com/support/softwaredownload/documents.aspx?standardid=18