UNH–IOL
NVMe Testing Service

Test Plan for NVMe
Computational Programs Command Set
Conformance
Version 25.0
Target Specification:
NVMe Computational Programs
Command Set Specification 1.0
Technical Document

University of
New Hampshire
Interoperability Labs

Last Updated: February 5th, 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

TABLE OF CONTENTS **2**

MODIFICATION RECORD **4**

ACKNOWLEDGMENTS **5**

INTRODUCTION **6**

REFERENCES **8**

ABBREVIATIONS **9**

Group 1: Admin Command Set **10**

Test 1.1 – Identify Command (FYI) **11**

Case 1: CNS=00h I/O Command Set Specific Controller, Version (FYI, OF- FYI) 11

Case 2: I/O Command Set Specific Identify Namespace, MRSG >= NOWG (FYI, OF- FYI) 11

Test 1.2 – Flush Command (FYI) **13**

Case 1: Flush Command, Compute Namespace (FYI) 13

Test 1.3 – Format Command (FYI) **14**

Case 1: Format Command, Compute Namespace (FYI, OF- FYI) 14

Test 1.4 – Sanitize Operations (FYI) **15**

Case 1: Sanitize Operation, Clear Downloaded Programs and Activation Settings (FYI) 15

Case 2: Sanitize Operation, No Effect on Device Defined Program's Activation Settings (FYI) 16

Case 3: Sanitize Command, No Effect on Memory Range Sets (FYI) 16

Case 4: Prohibited Commands Aborted when Sanitize In Progress SPROG (FYI) 16

Test 1.5 – Admin Log Pages (FYI) **18**

Case 1: Program List, PEOCC (FYI) 18

Case 2: Program List, IOS Field (FYI) 18

Case 3: Downloadable Program Types Log Page, VER Field (FYI) 19

Case 4: Memory Range Set List, Ascending RSID Order (FYI, OF- FYI) 19

Case 5: Memory Range Set List, MRS0 (FYI) 19

Case 6: Memory Range Set List, RIO Field (FYI) 19

Group 2: Computational Programs Command Set Specific Admin Commands **20**

Test 2.1 – Load Program Command (FYI) **21**

Case 1: Load Program Command, Program Index Not Downloadable (FYI) 21

Case 2: Load Program Command, PIND=FFFFh, SEL=0h (FYI) 21

Case 3: Load Program Command, No Program (FYI) 21

Case 4: Load Program Command, Reserved Program Type, Invalid Program Type (FYI) 22

Case 5: Load Program Command, NUMB, Invalid Field (FYI) 22

Case 6: Load Program Command, Unload All Programs (SEL=01b, PIND=FFFFh) (IP) 22

Test 2.2 – Memory Range Set Management (FYI) **23**

Case 1: Memory Range Set Management, Per-Namespace (FYI) 23

Case 2: Memory Range Set Management, Unique Non–Zero RSID (FYI) 23

Case 3: Memory Range Set Management, Delete (FYI) 24

Case 4: Memory Range Set Management, Delete, RSID=FFFFh (FYI) 24

Case 5: Memory Range Set Management, Invalid Memory Range Set (FYI) 24
Case 6: Memory Range Set Management, Invalid Memory Range Set Identifier (FYI) 25
Case 7: Memory Range Set Management, Invalid Memory Namespace (FYI) 25
Case 8: Memory Range Set Management, Memory Range Set In Use (FYI) 25
Case 9: Memory Range Set Management, Maximum Memory Ranges Exceeded (FYI) 26
Case 10: Memory Range Set Management, Maximum Memory Range Sets Exceeded (FYI) 26
Case 11: Memory Range Set Management, Overlapping Memory Ranges (FYI) 27
Test 2.3 – Program Activation Management (FYI) 28
Case 1: Program Activation Management, Activate and Deactivate (FYI) 28
Case 2: Program Activation Management, Per-Namespace Basis (FYI) 28
Case 3: Program Activation Management, Invalid Program Index (FYI) 29
Case 4: Program Activation Management, No Program (FYI) 29
Case 5: Program Activation Management, SEL = 0h, PIND=FFFFh (FYI) 29
Case 6: Program Activation Management, Activate with PIND=FFFFh (FYI) 30
Case 7: Program Activation Management, Program In Use (FYI) 30
Case 8: Program Activation Management, MAXACT (FYI) 30
Group 3: Computational Programs Command Set I/O Commands 32
Test 3.1 – Execute Program Command (FYI) 33
Case 1: Execute Command, Non-Zero RSID, Non-Zero NUMR (FYI) 33
Case 2: Execute Command, Invalid Data Length (FYI) 33
Case 3: Execute Command, Invalid Memory Namespace, Non-Existent Memory Namespace Identifier (FYI) 34
Case 4: Execute Command, Invalid Memory Namespace, Not Reachable (FYI) 34
Case 5: Execute Command, Program Not Activated (FYI) 35
Case 6: Execute Command, Invalid Memory Range Set Identifier (FYI) 35
Case 7: Execute Command, Invalid Program Index (FYI) 36
Case 8: Execute Command, Overlapping Memory Ranges (FYI) 36
Case 9: Execute Command, No Program (FYI) 36
Case 10: Execute Command, Maximum Memory Ranges Exceeded (FYI) 37
Group 4: Namespace Management Commands 38
Test 4.1 – Namespace Management (FYI) 39
Case 1: Namespace Management, Invalid Field in Command (FYI) 39
Test 4.2 – Namespace Attachment (FYI) 40
Case 1: Namespace Attachment, CSI=04h (FYI) 40
Appendix A: Default Test Setup 41
Appendix B: Notes on Test Procedures 42
Appendix C: Test Tools 43
Appendix D: NVMe Integrators List Requirements 44

MODIFICATION RECORD

2024 July 8 21 (Version 1.0) Initial Release

Carter Snay:

1. Initial release of the NVMe Computational Programs Command Set Conformance Test Suite with v22 of the NVMe ICC test plans.

2025 January 23 (Version 23.0)

Carter Snay:

1. Program revision update

2025 July 31 (Version 24.0)

Carter Snay:

1. Program revision update

2026 February 5 (Version 25.0)

Carter Snay:

1. Program revision update

ACKNOWLEDGMENTS

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

Carter Snay	UNH InterOperability Laboratory
Michael Crawley	UNH InterOperability Laboratory
Liam Warren	UNH InterOperability Laboratory

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 functionality of their products. This test suite is aimed at validating products in support of the work being directed by the NVMe Promoters Group.

These tests are designed to determine if a product conforms to specifications defined in the NVM Express Computational Programs Specification Revision 1.0 specification and NVM Express Base Specification 2.0d. Successful completion of these tests provides a reasonable level of confidence that the Device Under Test (DUT) will function properly in many NVMe environments.

The tests contained in this document are organized to simplify the identification of information related to a test, and to facilitate the actual testing process. Tests are separated into groups, primarily to reduce setup time in the lab environment, however the different groups typically also tend to focus on specific aspects of device functionality. A two–number, dot–notated naming system is used to catalog the tests. This format allows for the addition of future tests in the appropriate groups without requiring the renumbering of the subsequent tests.

The test definitions themselves are intended to provide a high–level description of the motivation, resources, procedures, and methodologies specific to each test. Formally, each test description contains the following sections:

Purpose
The purpose is a brief statement outlining what the test attempts to achieve. The test is written at the functional level.

References
This section specifies all reference material *external* to the test suite, including the specific references for the test in question, and any other references that might be helpful in understanding the test methodology and/or test results. External sources are always referenced by a bracketed number (e.g., [1]) when mentioned in the test description. Any other references in the test description that are not indicated in this manner refer to elements within the test suite document itself (e.g., "Appendix 5.A", or "Table 5.1.1–1").

Resource Requirements
The requirements section specifies the test hardware and/or software needed to perform the test. This is generally expressed in terms of minimum requirements, however in some cases specific equipment manufacturer/model information may be provided.

Last Modification
This specifies the date of the last modification to this test.

Discussion
The discussion covers the assumptions made in the design or implementation of the test, as well as known limitations. Other items specific to the test are covered here as well.

Test Setup
The setup section describes the initial configuration of the test environment. Small changes in the configuration should not be included here and are generally covered in the test procedure section (next).

Procedure

The procedure section of the test description contains the systematic instructions for carrying out the test. It provides a cookbook approach to testing, and may be interspersed with observable results. These procedures should be the ideal test methodology, independent of specific tool limitations or restrictions.

Observable Results
This section lists the specific observable items that can be examined by the tester in order to verify that the DUT is operating properly. When multiple values for an observable are possible, this section provides a short discussion on how to interpret them. The determination of a pass or fail outcome for a particular test is generally based on the successful (or unsuccessful) detection of a specific observable.

Possible Problems
This section contains a description of known issues with the test procedure, which may affect test results in certain situations. It may also refer the reader to test suite appendices and/or other external sources that may provide more detail regarding these issues.

REFERENCES

The following documents are referenced in this text:

1. NVM Express Computational Programs Command Set Specification Revision 1.0 (December 20, 2023)
2. NVM Express Base Specification Revision 2.0d (January 11, 2024)

ABBREVIATIONS

The following abbreviations are applied to the test titles of each of the tests described in this document for indicating the status of test requirements.

M - Mandatory
FYI - FYI
IP - In Progress

The following abbreviations applied to the test titles of each of the tests described in this document for indicating what product types a test may apply to. It is assumed that all tests apply to base NVMe products using PCIe.

OF – Test applies to NVMe-oF products

Group 1: Admin Command Set

Overview:

This section describes a method for performing conformance verification for NVMe products implementing the Admin Command Set.

Notes:

The preliminary draft descriptions for the tests defined in this group are considered complete, and the tests are pending implementation (during which time additional revisions/modifications are likely to occur).

Test 1.1 – Identify Command (FYI)

Purpose: To verify that an NVMe Controller can properly execute an Identify command.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.1.5

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Discussion: The Identify command returns a data buffer that describes the NVM subsystem, the controller or the namespace(s). The data structure is 4096 bytes in size. The host indicates as a command parameter whether to return the controller or namespace specific data structure. For the namespace data structure, the definition of the structure is specific to the I/O command set selected for use.

The data structure returned is based on the Controller or Namespace Structure (CNS) field. If there are fewer namespace identifiers or controller identifiers to return for a Namespace List or Controller List, respectively, then the unused portion of the list is zero filled. Controllers that support Namespace Management shall support CNS values of 10h–13h.

The Identify command uses the PRP Entry 1, PRP Entry 2, and Command Dword 10 fields. All other Command specific fields are reserved. A completion queue entry is posted to the Admin Completion Queue when the Identify data structure has been posted to the memory buffer indicated in PRP Entry 1.

Test Setup: See Appendix A.

Case 1: CNS=00h I/O Command Set Specific Controller, Version (FYI, OF- FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command for the Memory I/O Command Set Specific Controller Data Structure (CNS=06h, CSI=04h).

Observable Results:

1. Verify the identify command completed successfully.
2. Verify that the Specification Version Descriptor reports values of 1h in the MJR field, and 0h in the MNR and TER fields

Case 2: I/O Command Set Specific Identify Namespace, MRSG >= NOWG (FYI, OF- FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to obtain the I/O Command Set specific Identify Namespace data structure (CNS 05h, CSI 03h) for the Subsystem Local Memory Command Set. Record the NOWG.
2. Configure the NVMe Host to issue an Identify command to obtain the I/O Command Set specific Identify Namespace data structure (CNS 05h, CSI 04h) for the Computational Programs Command Set. Record the MRSG field.

Observable Results:

1. Verify the MRSG field is greater than or equal to NOWG.
2. Verify both Identify commands completed successfully.

Test 1.2 – Flush Command (FYI)

Purpose: To verify that an NVMe Controller performs the proper actions when a Flush Command occurs.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 2.2

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Discussion: A Flush command is expected to have no impact on compute namespaces. This test will verify that the controller will post a completion queue entry for the Flush command that does not result in an error status.

Test Setup: See Appendix A.

Case 1: Flush Command, Compute Namespace (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Flush command to a compute namespace.

Observable Results:

1. Verify the Flush command completed successfully.

Test 1.3 – Format Command (FYI)

Purpose: To verify that an NVMe Controller performs the proper actions when a Format Command Occurs.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.1.2

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8 2024

Discussion: The Format NVM command is not supported by the Computational Programs I/O Command Set Specification. When a Format NVM command is directed at a compute namespace, the controller is required to return an error status of "Invalid Namespace or Format"

Test Setup: See Appendix A.

Case 1: Format Command, Compute Namespace (FYI, OF- FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Format command specifying a Compute Namespace Identifier to attempt to format a compute namespace to specify FLBAS of 0.

Observable Results:

1. Verify that the Format command returned with status 'Invalid Namespace or Format'.

Test 1.4 – Sanitize Operations (FYI)

Purpose: To verify that an NVMe Controller can properly perform a Sanitize operation with the Computational Programs Command Set enabled.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.1.8

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Test Setup: See Appendix A.

Discussion: The Sanitize command is used to start a sanitize operation or to recover from a previously failed sanitize operation. The sanitize operation types that may be supported are Block Erase, Crypto Erase, and Overwrite. All sanitize operations are processed in the background (i.e., completion of the Sanitize command does not indicate completion of the sanitize operation)

For the Computational Programs Command Set, the Sanitize command deletes all downloaded programs and removes all associated PIND entries, as well as any program related remnants or parameters that may be in the internal compute environment. The Sanitize command does not affect device-defined programs.

Case 1: Sanitize Operation, Clear Downloaded Programs and Activation Settings (FYI)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the PEOCC field and 01b in the Activation field.
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field:
        i. Store the index of a Program Descriptor which has a value of 01b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b in the PEOCC, this test is not applicable.
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
3. Configure the NVMe Host to issue a Sanitize command specifying any supported sanitize action.
4. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.

Observable Results:

1. Verify that all commands completed successfully.
2. Verify that in the Get Log Page command from step 4 there are no descriptors in the Program list which have a PEOCC field set to 01b.

Case 2: Sanitize Operation, No Effect on Device Defined Program's Activation Settings (FYI)

Test Procedure:

1. Check SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.
    a. Store the index of a Program descriptor from the Program List log page with a value of 10b in the PEOCC field and 01b in the Activation field..
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field:
        i. Store the index of a Program Descriptor which has a value of 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 10b in the Program List Log Page, this test is not applicable.
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
3. Configure the NVMe Host to issue a Sanitize command specifying any supported Sanitize Action.
4. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.

Observable Results:

1. Verify that the commands in steps 1 and 2 were successful.
2. Verify that the Sanitize command had no effect on the device defined programs returned in the Program List from step 4.

Case 3: Sanitize Command, No Effect on Memory Range Sets (FYI)

Test Procedure:

1. Check the SANICAP field in the Identify Controller Data Structure. If SANICAP is set to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Memory Range Set Management command with SEL=0h to create a Memory Range Set on a Memory namespace.
3. Configure the NVMe Host to issue a Get Log Page command for the Memory Range Set List (LID 84h).
4. Configure the NVMe Host to issue an Sanitize command targeting the current Compute namespace.
5. Configure the NVMe Host to issue a second Get Log Page command for the Memory Range Set List (LID 84h).

Observable Results:

1. Verify all commands completed successfully.
2. Verify that both log pages returned in step 3 and 5 contain identical Memory Range Set Entries.

Case 4: Prohibited Commands Aborted when Sanitize In Progress SPROG (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command specifying a CNS value of 01h for Identify Controller. If the SANICAP field is set to 0, this test is not applicable.
2. Perform a Sanitize command, before command completion arrives for the Sanitize command, issue the following commands:
    a. Execute Program
    b. Memory Range Set Management
    c. Program Activation

Observable Results:

1. Verify that each command is aborted with a status of Sanitize In Progress.

Test 1.5 – Admin Log Pages (FYI)

Purpose: To verify that an NVMe Controller can properly execute a Get Log Page commands specific to Computational Programs.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.1.4

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Test Setup: See Appendix A.

Discussion: The Get Log Page command returns a data buffer that contains the log page requested. The Get Log Page command uses the PRP Entry 1, PRP Entry 2, and Command Dword 10 fields. All other command specific fields are reserved

The desired log page is specified in the Log Page Identifier (LID) field of CDW10. Valid Log Identifiers for the Computational Programs Command Set are described in Table 1

Table 1 – NVM Command Set Specific Log Page Identifiers

Log Identifier	Scope	Log Page Name
82h	Namespace	Program List
83h	Namespace	Downloadable Program Types List
84h	Namespace	Memory Range Set List

Case 1: Program List, PEOCC (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log Page command for the Program List (LID 82h). If no Programs are reported this test is not applicable.

Observable Results:

1. Verify that a Program Descriptor entry which reports a value of 0h for the PEOCC field reports a value of 0h in the PIT, Activation, PTYPE, and PID fields.
2. Verify that a Program Descriptor entry which reports a value greater than 0h for the PEOCC field reports a non-zero value for the PIT, and PID fields

Case 2: Program List, IOS Field (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command for LID 00h, Supported Log Pages. If the IOS field of the LID Supported and Effects Data Structure for LID 82h is cleared to 0, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log Page command for the Program List (LID 82h) clearing the IOS field to 0h. If no Programs are reported this test is not applicable.

3. Configure the NVMe Host to issue a Get Log Page command for the Program List (LID 82h) setting the IOS field to 1h.

Observable Results:
1. Verify all Get Log Page commands completed successfully.
2. Verify that the Get Log Page command in step 2 returns all programs starting at index 0.
3. Verify that the Get Log Page command in step 3 returns all programs starting at index 1.

Case 3: Downloadable Program Types Log Page, VER Field (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log Page command for the Downloadable Program Types List (LID 83h). If no Program Types are reported this test is not applicable.

Observable Results:
1. Verify that all entries in the Downloadable Program Types List report a value of 0h in the VER field.

Case 4: Memory Range Set List, Ascending RSID Order (FYI, OF- FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log Page command for the Memory Range Set List (LID 84h).

Observable Results:
1. Verify that all entries are returned in ascending order according to the value in their RSID fields

Case 5: Memory Range Set List, MRS0 (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log Page command for the Memory Range Set List (LID 84h).

Observable Results:
1. Verify that the value of NUMD reported in the Memory Range Set List is not cleared to 0.
2. Verify that there is an entry for MRS0 in the Memory Range Set List

Case 6: Memory Range Set List, RIO Field (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log Page command for the Memory Range Set List (LID 84h) with the RIO field set to 1

Observable Results:
1. Verify the Get Log Page command completed successfully.
2. Verify that each Memory range set returned in the Get Log Page command has its NMR field cleared to 0h.

Group 2: Computational Programs Command Set Specific Admin Commands

Test 2.1 – Load Program Command (FYI)

Purpose: To verify that an NVMe Controller is able to successfully load a computational program, and return expected statuses in scenarios where the programs are invalid or non-existent.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 :4.2.1

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Test Setup: See Appendix A.

Discussion: The Load Program command is used to load downloadable programs, which will be required if supported to retrieve externally defined programs onto a device. Supported program types can be discovered in the Downloadable Program Types Log Page [1].

Case 1: Load Program Command, Program Index Not Downloadable (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List, and record the index of a program which reports a value of 10b in the PEOCC field. If there are no device-defined programs this test is not applicable.
2. Configure the NVMe Host to issue a Load Program command with SEL=1b and PIND set to the index of a device-defined program.
3. Configure the NVMe Host to issue a Load Program command with SEL=0b and PIND set to the index of a device-defined program.

Observable Results:

1. Verify the Get Log command in step 1 completed successfully.
2. Verify that both Load Program commands in step 2 and 3 return a status of ‘Program Index Not Downloadable’.

Possible Problems: None.

Case 2: Load Program Command, PIND=FFFFh, SEL=0h (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Load Program command with a PIND value of FFFFh and SEL=0b.

Observable Results:

1. Verify that the Load Program command completed with the status ‘Invalid Program Index’

Case 3: Load Program Command, No Program (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List, and record the index of a program which reports a value of 00b in the PEOCC field.
2. Configure the NVMe Host to issue a Load Program command with SEL=1b for unload, and a PIND that is unoccupied.

Observable Results:
1. Verify the Get Log command in step 1 was Successful.
2. Verify that the Load Program command completes with the status 'No Program'.

Case 4: Load Program Command, Reserved Program Type, Invalid Program Type (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Load Program command with SEL=0b and a PIT value of 7Fh.

Observable Results:
1. Verify that the Load Program command completes with the status 'Invalid Program Type'.

Case 5: Load Program Command, NUMB, Invalid Field (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Command Set Specific Identify Namespace command (CNS 05h CSI=04h) and store the value of the LPG field.
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h for Program List and store the Index of a Program Descriptor which reports a value of 00b in its PEOCC field. If no Program Descriptors report a value of 00b in the PEOCC field this test is not applicable.
3. Configure the NVMe Host to issue a Load Program command with NUMB=1h and PIND equal to the value recorded in step 2 .
4. Configure the NVMe Host to issue a Load Program command with a value for NUMB which is LPG^2 + 1.

Observable Results:
1. Verify the Identify command in step 1 completed successfully.
2. Verify that the Load Program command in steps 2 and 3 completed with the status 'Invalid Field'.

Case 6: Load Program Command, Unload All Programs (SEL=01b, PIND=FFFFh) (IP)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 82h for Program List. If no Program Descriptors report a value of 01b in the PEOCC field this test is not applicable
2. Configure the NVMe Host to issue a Load Program command with a PIND value of FFFFh and a SEL value of 1b.

Observable Results:
1. Verify that the Get Log command in step 1 and the Load Program command in step 2 completed successfully.
2. Verify that all programs reported in the Program List Log page report a value for PEOCC of 00b or 10b.

Test 2.2 – Memory Range Set Management (FYI)

Purpose: To verify an NVMe controller can properly create and delete Memory Range sets on a per-namespace basis

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.2.2

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification:

Discussion: Memory Range Set Management is a mandatory command, which is used to create and delete Memory Range sets to be used in computational operations. A memory range sets is one or more ranges of subsystem local memory, and they exist within a compute namespace. The purpose of creating Memory Range Sets is to limit program access to specific subsets of subsystem local memory.

Test Setup: See Appendix A.

Case 1: Memory Range Set Management, Per-Namespace (FYI)

Test Procedure:

1. Configure the NVMe host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure (CNS=05h, CSI=04h). Record the MRSG field.
2. Configure the NVMe host to issue an Identify command with CNS=07h and CSI=04h. If less than 2 namespaces are reported this test is not applicable.
3. Configure the NVMe host to issue a Memory Range Set Management command with an SEL value of 0h for create, a valid MNSID value, and a Range Descriptor with a Length field that is a multiple of 2 ^ MRSG.
4. For each Compute namespace reported in step 2, configure the NVMe host to issue a Get Log command with a LID of 84h for the Memory Range Set List log.

Observable Results:

1. Verify the Identify command in step 1 completed successfully.
2. Verify the Memory Range Set Management command in step 3 completed successfully.
3. Verify the Memory Range Set created in step 3 is only reported in the Memory Range Set list of the namespace it was created on.

Case 2: Memory Range Set Management, Unique Non–Zero RSID (FYI)

Test Procedure:

1. Configure the NVMe host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure (CNS=05h, CSI=04h). Record the MRSG field.
2. Configure the NVMe host to issue a Get Log command with a LID of 84h for the Memory Range Set List.
3. Configure the NVMe host to issue a Memory Range Set Management command with SEL=0b for create, and a DPTR specifying a Memory Range Descriptor with a Length field that is a multiple of 2 ^ MRSG.
4. Configure the NVMe host to issue a Get Log command with a LID of 84h for the Memory Range Set List.

Observable Results:

1. Verify all commands completed successfully.
2. Verify a new Memory Range Set was created in step 3 and a unique non–zero RSID value is returned in Dword 0.

Case 3: Memory Range Set Management, Delete (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Memory Range Set List log page (LID 84h).
    a. If no Memory Range Sets other than RSID 0 are reported, configure the NVMe host to create a Memory Range Set.
2. Configure the NVMe host to issue a Memory Range Set Management command with an SEL= 01b for delete, and a valid non-zero RSID.
3. Configure the NVMe host to issue a second Memory Range Set List log page (LID 84h).

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Memory Range Set Management command in step 1 completed successfully
3. Verify that the Memory Range Set List log page does not report the Memory Range Set that was deleted in step 2.

Case 4: Memory Range Set Management, Delete, RSID=FFFFh (FYI)

Test Procedure:

1. Configure the NVMe host to issue a second Memory Range Set List log page (LID 84h).
    a. If no Memory Range Sets other than RSID 0 are reported, configure the NVMe host to create 2 Memory Range Sets.
2. Configure the NVMe host to issue a Memory Range Set Management command with an SEL value of 01b for delete, and an RSID value of FFFFh.
3. Configure the NVMe host to issue a second Memory Range Set List log page (LID 84h).

Observable Results:

1. Verify all commands completed successfully.
2. Verify the Memory Range Set Management command with RSID=FFFFh deleted all range sets with non-zero RSID values.

Case 5: Memory Range Set Management, Invalid Memory Range Set (FYI)

Test Procedure:

1. Configure the NVMe host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure (CNS=05h, CSI=04h). Record the MRSG field.
2. Configure the NVMe host to issue a Memory Range Set Management command with a SEL value of 0h for create and a data buffer specifying a length that is less than 2 ^ MRSG.

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify the Memory Range Set Management command in step 2 returned a status of “Invalid Memory Range Set.”

Case 6: Memory Range Set Management, Invalid Memory Range Set Identifier (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with a LID of 84h (Memory Range Set List) and store a value which does not correspond to any of the reported RSID's.
2. Configure the NVMe Host to issue a Memory Range Set Management command with a SEL value of 1h for Delete and the RSID value that was recorded in step 1.

Observable Results:

1. Verify that the Get Log command in step 1 completed successfully.
2. Verify that the Memory Range Set Management command in step 1 completed with the status "Invalid Memory Range Set Identifier."

Case 7: Memory Range Set Management, Invalid Memory Namespace (FYI)

1. Check the Reachability Associations and Groups Log Pages to determine:
    a. If there is a memory namespace which is not reachable to the compute namespace. If there are no unreachable namespaces skip step 2.
    b. If there is a non-memory namespace which is reachable to the compute namespace. If there is no non memory namespace which is reachable to the compute namespace, skip step 5.
2. Configure the NVMe host to issue a Memory Range Set Management command with a SEL value of 0h for create and a data buffer specifying a Memory Range with a MNSID field of the memory namespace determined in step 1a..
3. Configure the NVMe host to issue a Memory Range Set Management command with a SEL value of 0h for create and a data buffer containing an MNSID field of 0h.
4. Configure the NVMe host to issue a Memory Range Set Management command with a SEL value of 0h for create and a data buffer containing an MNSID field of FFFFFFFFh.
5. Configure the NVMe host to issue a Memory Range Set Management command with a SEL value of 0h for create and a data buffer specifying a Memory Range with a MNSID field of the namespace determined in step 1b.

Observable Results:

1. Verify the Get Log command in step 1 completed successfully.
2. Verify all Memory Range Set Management commands complete with the status "Invalid Memory Namespace."

Case 8: Memory Range Set Management, Memory Range Set In Use (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Memory Range Set List log page (LID 84h).
    a. If no Memory Range Sets other than RSID 0 are reported, configure the NVMe host to create a Memory Range Set.
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field.
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field:

   i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
   ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
3. Configure the NVMe host to issue an Execute Program command specifying a PIND value of that stored in step 2, and a RSID of the Memory Range Set recorded in step 2.
4. Before the Completion Queue entry for the Execute Program command in step 1 is posted. Configure the NVMe host to issue a Memory Range Set Management command with a SEL value of 1h for delete and a RSID value of that specified in the Execute Program command in step 1.

Observable Results:
1. Verify the Memory Range Set Management command in step 1 returned a status of "Memory Range Set In Use."
2. Verify all other commands completed successfully.

Possible Problems: The controller may process the Execute Program command before the host can issue the commands in step 4. In this case the commands should complete successfully.

Case 9: Memory Range Set Management, Maximum Memory Ranges Exceeded (FYI)

Test Procedure:
1. Configure the NVMe host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure (CNS=05h, CSI=04h). Record the MRSG and MAXMEMR fields. If the DUT reports a value of 0h in this field, this test is not applicable.
2. Configure the NVMe Host to issue a Memory Range Set Management command specifying an SEL value of 0h for create, and a value for NUMR greater than the value recorded in step 1 and less than 128.
   a. If the DUT reported a value greater than 128 in the MAXMEMR field, repeat this step until the total number of Memory Ranges would exceed the value in the MAXMEMR field.

Observable Results:
1. Verify that the Identify command and all Memory Range Set Management commands that do not exceed MAXMEMR completed successfully.
2. Verify that the Memory Range Set Management command in step 2 causes the total number of Memory Ranges to exceed MAXMEMR completed with status 'Maximum Memory Ranges Exceeded'.

Case 10: Memory Range Set Management, Maximum Memory Range Sets Exceeded (FYI)

Test Procedure:
1. Configure the NVMe host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure (CNS=05h, CSI=04h). Record the MRSG and MAXMEMRS fields. If the MAXMEMRS is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue MAXMEMRS Memory Range Sets Management commands with SEL=0h to create the maximum number of Memory Range Sets.
3. Configure the NVMe Host to issue a Memory Range Sets Management commands with SEL=0h to attempt to create an additional Memory Range Set.

Observable Results:
1. Verify that the Identify command issued in step 1 completed successfully.

2. Verify that all the Memory Range Set Management commands in step 2 completed successfully.
3. Verify that the Memory Range Set Management command in step 3 completed with status 'Maximum Range Sets Exceeded'.

Case 11: Memory Range Set Management, Overlapping Memory Ranges (FYI)

Test Procedure:

1. Configure the NVMe host to issue an Identify command to obtain the I/O Command Set Specific Identify Namespace data structure (CNS=05h, CSI=04h). Record the MRSG field.
2. Configure the NVMe Host to issue a Memory Range Set Management command specifying an SEL value of 0h for Create and a data pointer specifying a Memory Range Descriptor with a valid starting byte and length, and a second Memory Range Descriptor with a starting byte within the range of the first Memory Range Descriptor and a valid length.

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify that the Memory Range Set Management command issued in step 1 returned a status of "Overlapping Memory Ranges".

Test 2.3 – Program Activation Management (FYI)

Purpose: To verify an NVMe controller can properly activate and deactivate programs on a per-namespace basis

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.2.3

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Discussion: The Program Activation Management command is a mandatory command which activates and deactivates programs. This command takes the Program Index (PIND) and action of either Activate or Deactivate, which is used to specify which programs on a device are active at any given time. The state of a program (active or inactive) is not persistent across power cycles.

Test Setup: See Appendix A.

Case 1: Program Activation Management, Activate and Deactivate (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h and record the index of a Program Descriptor which reports a value of 0b in its Activation field.
2. Configure the NVMe Host to issue a Program Activation Management command specifying an SEL value of 01h for Activate and a PIND value specifying the Program Index of a Program which is not activated that was found in step 1.
3. Configure the NVMe Host to issue a second Get Log command with an LID of 82h.
4. Configure the NVMe Host to issue a Program Activation Management command specifying an SEL value of 00h for Deactivate and a PIND value the same as the PIND specified in step 2.
5. Configure the NVMe Host to issue a third Get Log command with an LID of 82h.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the index which was specified in the Program Activation Management command in step 2 reports a value of 01b in the Activation field of its entry in the Program List Log page command issued in step 3.
3. Verify that the index which was specified in the Program Activation Management command in step 4 reports a value of 00b in the Activation field of its entry in the second Program List Log Page command issued in step 5.

Case 2: Program Activation Management, Per-Namespace Basis (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command with CNS 07h, and CSI 04h. Record the NSID of 2 compute namespaces. If less than 2 compute namespaces are reported this test is not applicable.

2. Configure the NVMe Host to issue a Get Log command with an LID of 82h to the two compute namespaces recorded in step 1 and record the index of a Program Descriptor which reports a value of 0b in its activation field on both namespaces.
    a. If the compute namespaces do not a program which has an activation field cleared to 0h on both namespaces, configure the NVMe Host to issue one or more Program Activation Management commands to configure each Namespace such that there is a Program Index which reports a Activation value cleared to 0h on both Namespaces.
3. Configure the NVMe Host to issue a Program Activation Management command specifying an SEL value of 01h for activate and a PIND value of that recorded in step 2.
4. Configure the NVMe Host to issue a Get Log command with an LID of 82h to each compute namespace.

Observable Results:

1. Verify all commands completed successfully.
2. Verify that the namespace which was specified in the Program Activation Management command in step 2 reports a value of 1b in the Activation field of its entry in the Program List Log page.
3. Verify that the namespace which was not specified in the Program Activation Management command in step 2 reports a value of 0b in the Activation field of its entry in the Program List Log page.

Case 3: Program Activation Management, Invalid Program Index (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List Log, and store the value of the NUMD field.
2. Configure the NVMe Host to issue a Program Activation Management command specifying SEL=1b (Activate) and a PIND value greater than the value of the NUMD field reported in the Program List Log page.

Observable Results:

1. Verify the Get Log command completed successfully.
2. Verify that the Program Activation Management command returned a status of “Invalid Program Index”.

Case 4: Program Activation Management, No Program (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List Log, and store the Index of a program which reports a value cleared to 0b in its PEOCC field**.**
2. Configure the NVMe Host to issue a Program Activation Management command specifying SEL=01h (Activate) and a PIND value of that stored in step 1.

Observable Results:

1. Verify that the Get Log command in step 1 was completed successfully.
2. Verify that the Program Activation Management command returned a status of “No Program”.

Case 5: Program Activation Management, SEL = 0h, PIND=FFFFh (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List log page, and store the Index of a program which reports a value of 01b in its PEOCC field**.**
2. Configure the NVMe Host to issue a Program Activation Management command specifying an SEL value of 0h for deactivate and a PIND value of FFFFh to deactivate all programs.

3. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List log page**.**
4. Configure the NVMe Host to issue a Program Activation Management command specifying an SEL value of 0h for deactivate and a PIND value of FFFFh to deactivate all programs.

Observable Results:
1. Verify all commands completed successfully.
2. Verify that the Program List log page reported all Program Descriptors with the Activation field cleared to 0.

Case 6: Program Activation Management, Activate with PIND=FFFFh (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Program Activation Management command specifying SEL=01h (Activate) and a PIND value of FFFFh.

Observable Results:
1. Verify that the Program Activation Management command returned a status of "Invalid Program Index".

Case 7: Program Activation Management, Program in Use (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field.
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field:
        i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
2. Configure the NVMe Host to issue an Execute Program command specifying the PIND that was recorded in step 1.
3. Before the Completion Queue Entry for the Execute Program command is posted to the Completion Queue, configure the NVMe Host to issue a Program Activation Management command specifying an SEL value of 0h for Deactivate and a PIND of the program that was specified in the Execute command in step 1.

Observable Results:
1. Verify that the Execute Program Command completed successfully.
2. Verify that the Program Activation Management Commands returned a status of "Program In Use".

Possible Problems: It is possible for the controller to wait and process the Execute Program command before sending the Program Activation Management command. If the Program Activation Management command completed successfully, verify that the Program Index of the Program specified reports a value cleared to 0h in its activation field.

Case 8: Program Activation Management, MAXACT (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an I/O Command Set Specific Identify Namespace Command (CNS 05h, CSI 04h). If a value of 0h is reported in the MAXACT field this test is not applicable.
2. Configure the NVMe Host to issue a Get Log command with LID=82h.If there are less than MAXACT+1 program indices occupied; this test is not applicable.
3. Configure the NVMe Host to issue a Program Activation Management command specifying a value of 01h for activate and a valid PIND. Repeat this step until the total number of active programs equal to MAXACT.
4. Configure the NVMe Host to issue a Program Activation Management command specifying a value of 01h for activation and a valid PIND to exceed MAXACT.

Observable Results:

1. Verify that the Program Activation Management command in step 4 completed with the status “Maximum Programs Activated”.
2. Verify that all other commands are completed successfully.

Group 3: Computational Programs Command Set I/O Commands

Test 3.1 – Execute Program Command (FYI)

Purpose: To verify that an NVMe controller can execute programs which are specified in this command, along with the data and parameters in the SLM.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 3.2.1

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Discussion: The Execute Program command is used to execute the program selected based on the PIND, data in the specified Memory Range Sets, and parameters in the Execute Program Command.

Test Setup: See Appendix A.

Case 1: Execute Command, Non-Zero RSID, Non-Zero NUMR (FYI)

Test Procedure:

1. Configure the NVMe host to issue a Get Log command with an LID of 84h, Memory Range Set List, and record the RSID of a Memory Range Set other than Memory Range Set 0
    a. If no Memory Range Sets other than Memory Range Set 0 are reported in the Memory Range Set List, Configure the NVMe host to issue a Memory Range Set Management command specifying an SEL value of 0h for Create, and record the RSID returned by the command.
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List.
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field.
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field:
        i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
3. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND that was recorded in step 2, an RSID that was recorded in step 1, and the NUMR field and DPTR fields describing valid Memory Ranges.

Observable Results:

1. Verify the Execute Program command completed with the status “Invalid Field in Command”.
2. Verify all other commands completed successfully.

Case 2: Execute Command, Invalid Data Length (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List

a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field.
b. If no Program Descriptor in the Program List has a value of 01b for the Activation field,
    i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
    ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.

2. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 1, and the DLEN field set to a value less than NUMR * 32.

Observable Results:

1. Verify the Execute Program command completed with status “Invalid Field in Command”.

Case 3: Execute Command, Invalid Memory Namespace, Non-Existent Memory Namespace Identifier (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command with a CNS value of 07h for the I/O Command Set specific Active Namespace ID list, and a CSI value of 03h. Store a value which does not correspond to any of the NSIDs returned.
2. Configure the NVMe Host to issue an Identify command with a CNS value of 07h for the I/O Command Set specific Active Namespace ID list, and a CSI value of 00h. Store one of the NSIDs returned.
3. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field.
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field,
        i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
4. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND that was recorded in step 2, an RSID value cleared to 0, and a DPTR specifying memory ranges with a Memory Namespace field that was recorded in step 1.
5. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that was recorded in step 2, an RSID value cleared to 0, and a DPTR specifying memory ranges with a Memory Namespace field that was recorded in step 2. If no namespace were returned in the Identify Command in step 2, this step is not applicable.

Observable Results:

1. Verify both Execute Program commands completed with status “Invalid Memory Namespace”.
2. Verify all other commands completed successfully.

Case 4: Execute Command, Invalid Memory Namespace, Not Reachable (FYI)

Test Procedure:

1. Check the Reachability Associations and Reachability Groups log pages to determine if there is a memory namespace which does not share a reachability association with the compute namespace, if no memory namespaces are present which are not reachable this test is not applicable.
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List

   a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field
   b. If no Program Descriptor in the Program List has a value of 01b for the Activation field,
      i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
      ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
3. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 2, an RSID value cleared to 0, and a DPTR specifying a Memory Range with an MNSID of that recorded in step 1.

Observable Results:
1. Verify the Execute Program command completed with status "Invalid Memory Namespace".
2. Verify all commands completed successfully.

Case 5: Execute Command, Program Not Activated (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List
   a. Store the index of a Program descriptor from the Program List log page with a value cleared to 0b in the Activation field.
   b. If no Program Descriptor in the Program List has a value cleared to 0b for the Activation field:
      i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable
      ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 0h and a PIND value of the stored Program Descriptor.
2. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 1, and a DPTR specifying a valid Memory Range Set.

Observable Results:
1. Verify the Execute Program command completed with status "Program Not Activated".
2. Verify all other commands completed successfully

Case 6: Execute Command, Invalid Memory Range Set Identifier (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 84h, Memory Range Set List, and store the RSID field of the last Memory Range Set Descriptor (Memory Range Set NUMD-1).
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List
   a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field
   b. If no Program Descriptor in the Program List has a value of 01b for the Activation field,
      i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a PEOCC value of 01b or 10b in the Program List Log Page, this test is not applicable.
      ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.

3. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 2, and the RSID field set to the value recorded in step 1 plus 1.

Observable Results:
1. Verify the Execute Program command completed with status “Invalid Memory Range Set Identifier”.
2. Verify all other commands completed successfully.

Case 7: Execute Command, Invalid Program Index (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List, and record the value of the NUMD field.
2. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 1 plus 1 and a non-zero NUMR value and a DPTR specifying valid memory ranges.

Observable Results:
1. Verify the Get Log Page command in step 1 completed successfully.
2. Verify the Execute Program command completed with status “Invalid Program Index”.

Case 8: Execute Command, Overlapping Memory Ranges (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field,
        i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable.
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
2. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 1, and a Data Buffer with fields specifying one valid Memory Ranges, and one memory range which has a starting byte within the bytes specified in the first Memory Range.

Observable Results:
1. Verify the Execute Program command completed with status “Overlapping Memory Ranges”.
2. Verify all other commands completed successfully.

Case 9: Execute Command, No Program (FYI)

Test Procedure:
1. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List and Store the index of a Program descriptor from the Program List log page with a value of 00b in the PEOCC field. If all PINDs are occupied this test is not applicable.
2. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 1, and a valid RSID value.

Observable Results:

1. Verify the Get Log Page command completed successfully.
2. Verify the Execute Program command completed with status “No Program”.

Case 10: Execute Command, Maximum Memory Ranges Exceeded (FYI)

Test Procedure:

1. Configure the NVMe Host to issue an Identify command to CNS=05h CSI=4h, Command Set Specific Identify Namespace Data Structure. If MAXMEMR is cleared to 0h, this test is not applicable.
2. Configure the NVMe Host to issue a Get Log command with an LID of 82h, Program List
    a. Store the index of a Program descriptor from the Program List log page with a value of 01b in the Activation field.
    b. If no Program Descriptor in the Program List has a value of 01b for the Activation field
        i. Store the index of a Program Descriptor which has a value of 01b or 10b in the PEOCC field. If the product under test has no Program Descriptors with a value of 01b or 10b in the Program List Log Page, this test is not applicable
        ii. Configure the NVMe Host to issue Program Activation Management Command with a SEL value of 01h and a PIND value of the stored Program Descriptor.
3. Configure the NVMe Host to issue an Execute Program command to a compute namespace with a value for PIND of that recorded in step 1, and a Data Buffer with fields specifying a number of Memory Ranges greater than the MAXMEMR field. Field.

Observable Results:

1. Verify the Execute Program command completed with status “Maximum Memory Ranges Exceeded”.
2. Verify all other commands completed successfully.

Group 4: Namespace Management Commands

Test 4.1 – Namespace Management (FYI)

Purpose: To verify that the Namespace Management command is not supported by a compute namespace.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.1.7

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Discussion: The Namespace Management command is not supported for compute namespaces. As a result, any Namespace Management command directed to a namespace that is a compute namespace, or uses the CSI value of 04h, the command is expected to abort with a status of "invalid Field in Command". It is expected that compute namespaces are configured at the time of firmware initialization, and are unable to be created or deleted.

Test Setup: See Appendix A.

Case 1: Namespace Management, Invalid Field in Command (FYI)

Test Procedure:

1. Configure the NVMe Host to issue a Namespace Management command with SEL=0h to attempt to create a namespace with a CSI value of 4h.
2. Configure the NVMe Host to issue a Namespace Management command with SEL=1h to attempt to delete a namespace with a CSI value of 4h.

Observable Results:

1. Verify both Namespace Management commands returned with a status code of 'Invalid Field in Command'

Test 4.2 – Namespace Attachment (FYI)

Purpose: To verify that an NVMe controller which supports the Namespace Attachment command operates in the same way that the NVMe Base specification [2] defines.

References:

[1] NVM Express Computational Programs Command Set Specification 1.0 : 4.1.6
[2] NVM Express Base Specification 2.0d : 5.22

Resource Requirements:

Tools capable of monitoring and decoding traffic on the NVMe interface.

Last Modification: July 8, 2024

Discussion: Compute namespaces cannot be deleted or created, but they can be attached and detached from an NVMe controller. This test verifies that the functionality of the Namespace Attachment command is the same as defined in the NVMe Base specification.

Test Setup: See Appendix A.

Case 1: Namespace Attachment, CSI=04h (FYI)

Test Procedure:

1. For each namespace in the NVM Subsystem, configure the NVMe Host to issue an Identify command for the I/O Command Set Specific Active Namespace list (CNS 7h) with a CSI value of 04h. Record the CSI value for each namespace if reported.
2. For each namespace which reports a CSI value of 4h:
    a. Configure the NVMe Host to issue a Namespace Attachment command with SEL=0h to detach the namespace from an attached controller.
    b. Configure the NVMe Host to issue an Namespace Attachment command with SEL=1h to attach the namespace back to the controller.

Observable Results:

1. Verify the Identify command completed successfully.
2. Verify the Attach and Detach Namespace commands completed successfully.

Case X: <Title Goes Here> (FYI)

Test Procedure:

1. Step 1
2. Step 2
3. Step 3

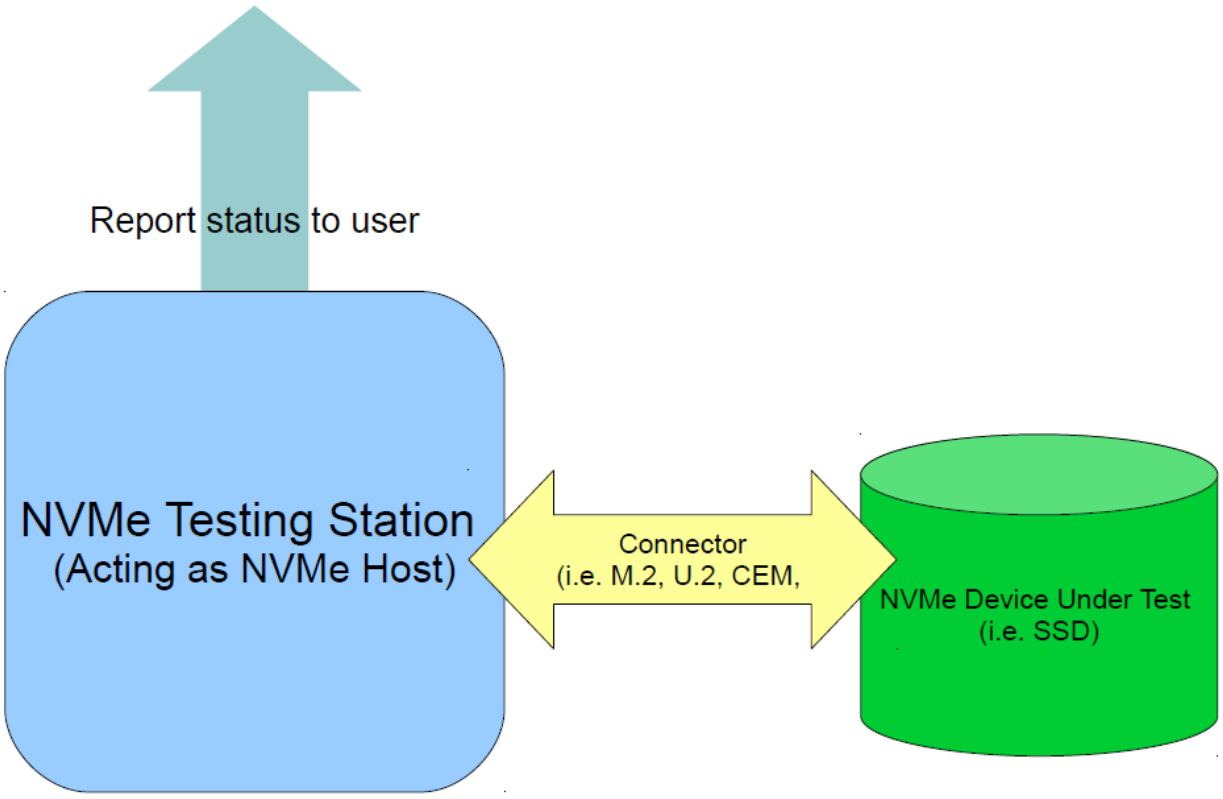
Observable Results:

1. Observable 1
2. Observable 2

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:

Test Setup for NVMe Device:



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 IOL INTERACT NVMe Test Software available from UNH–IOL.

If using the PC Edition of the IOL INTERACT NVMe Test Software, UNH-IOL recommends using v21.0 or higher of the IOL INTERACT NVMe Test Software. This software is available via https://www.iol.unh.edu/solutions/test-tools/interact.

Appendix D: NVME INTEGRATORS LIST REQUIREMENTS

Purpose: To provide guidance on what tests are required for NVMe Integrators List Qualification

References:

[1] NVMe Integrators List Policy Document

Resource Requirements:

NVMe Host Platform and Device.

Last Modification: April 2, 2019

Discussion: Each Test defined in this document is defined as being Mandatory (M), FYI, or In Progress (IP). This primary designation is shown in the title of the test case and is understood to apply to PCIe based products. An additional designation is provided if a test is applicable to NVMe-oF products (OF). Tests that are designated as being applicable to NVMe-oF Products are understood to inherit the primary designation of the test (i.e. M, FYI, IP), unless an additional designation is specified. The following examples are provided:

Test 1.1 Example Name (M)– Test is mandatory for all PCIe based products and does not apply to NVMe-oF products.
Test 2.1 Example Name (M, OF)– Test is mandatory for all products, including NVMe-oF products.
Test 3.1 Example Name (M, OF-IP)- Test is mandatory for all PCIe based products, and test is currently On Progress for NVMe-oF products.

NVMe protocol testing is independent of the transport used. Conformance tests described in this document may be performed at any link speed, width, or transport type that the NVMe product under test supports.

If a Test is designated as Mandatory, a product must pass this test in order to qualify for the NVMe Integrators List. For tests that deal with features defined as optional in the NVMe specification, a check is performed at the beginning of the test to determine if the optional feature is supported or not. If the optional feature is not supported the test is marked as 'Not Applicable' and does not impact qualification for the Integrators List.

If a Test is designated as FYI, a device does not need to pass this test in order to qualify for the NVMe Integrators List. Tests designated as FYI may become Mandatory tests in the future.

If a Test is designated as In Progress, a device does not need to pass this test in order to qualify for the NVMe Integrators List. These test cases are still under development. Tests designated as In Progress may become Mandatory tests in the future.

Any Test may have a Case within it with a different designation as the Test itself (i.e. a Mandatory test may include FYI cases). In this case, only the Mandatory Cases are required for NVMe Integrators List qualification.